AD-A075 679 SCIENCE APPLICATIONS INC ENGLEWOOD CO F/G 9/2

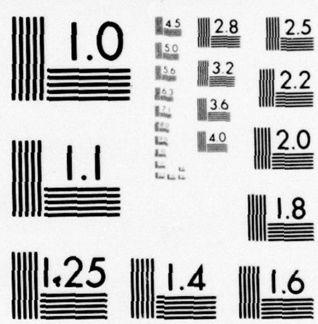
HUMAN FACTORS IN COMPUTER SYSTEMS: A REVIEW OF THE LITERATURE.(U)

SEP 79 H R RAMSEY , M E ATWOOD N00014-76-C-0866

UNCLASSIFIED SAI-79-111-DEN NL

1 OF 2

ADA 075679



MICROCOPY RESOLUTION TEST CHART

(12) LEVEL II

Technical Report SAI-79-111-DEN

HUMAN FACTORS IN COMPUTER SYSTEMS: A REVIEW OF THE LITERATURE

By

H. Rudy Ramsey
Michael E. Atwood

Science Applications, Inc.
7935 E. Prentice Ave.
Englewood, CO 80111

21 September 1979

This work was supported by the Engineering Psychology Programs, Office of Naval Research under Contract N00014-76-C-0866, Work Unit Number NR196-146.

Technical Report SAI-79-111-DEN

HUMAN FACTORS IN COMPUTER SYSTEMS: A REVIEW OF THE LITERATURE

By

H. Rudy Ramsey
Michael E. Atwood

Science Applications, Inc.
7935 E. Prentice Ave.
Englewood, CO 80111

21 September 1979

This work was supported by the Engineering Psychology Programs, Office of Naval Research under Contract N00014-76-C-0866, Work Unit Number NR196-146.

UNCLASSIFIED
SECURITY CLASSIFICATION OF THIS PAGE (When Data Entered)

REPORT DOCUMENTATION PAGE		READ INSTRUCTIONS BEFORE COMPLETING FORM
1. REPORT NUMBER	2. GOVT ACCESSION NO.	3. RECIPIENT'S CATALOG NUMBER 9
4. TITLE (and Subtitle) 6 HUMAN FACTORS IN COMPUTER SYSTEMS: A REVIEW OF THE LITERATURE.		5. TYPE OF REPORT & PERIOD COVERED Technical Report, 1 June 1976 — 31 July 1979
		6. PERFORMING ORG. REPORT NUMBER 14 SAI-79-111-DEN
7. AUTHOR(s) 10 H. Rudy Ramsey and Michael E. Atwood		8. CONTRACT OR GRANT NUMBER(s) 15 N00014-76-C-0866
9. PERFORMING ORGANIZATION NAME AND ADDRESS Science Applications, Inc. 7935 E. Prentice Avenue Englewood, Colorado 80111		10. PROGRAM ELEMENT, PROJECT, TASK AREA & WORK UNIT NUMBERS NR196-146
11. CONTROLLING OFFICE NAME AND ADDRESS Office of Naval Research 800 N. Quincy Street Arlington, Virginia 22217		12. REPORT DATE 11 21 September 1979
		13. NUMBER OF PAGES 169 pages
14. MONITORING AGENCY NAME & ADDRESS (if different from Controlling Office) 12 180		15. SECURITY CLASS. (of this report) UNCLASSIFIED
		15a. DECLASSIFICATION/DOWNGRADING SCHEDULE

16. DISTRIBUTION STATEMENT (of this Report)

Approved for public release; distribution unlimited.

17. DISTRIBUTION STATEMENT (of the abstract entered in Block 20, if different from Report)

18. SUPPLEMENTARY NOTES

19. KEY WORDS (Continue on reverse side if necessary and identify by block number)

Human Factors Engineering, Computers, Man Machine Systems, Interfaces, Interactions, Literature Surveys

20. ABSTRACT (Continue on reverse side if necessary and identify by block number)

See next page.

DD FORM 1 JAN 73 1473 EDITION OF 1 NOV 65 IS OBSOLETE

UNCLASSIFIED 392 878

SECURITY CLASSIFICATION OF THIS PAGE (When Data Entered)

UNCLASSIFIED
SECURITY CLASSIFICATION OF THIS PAGE(When Data Entered)

Based on an extensive literature survey, this document presents a description and critical analysis of the state of the art in the area of human factors in computer systems. This review is concerned both with the status of human factors research in the area of user computer interaction and with the current state of user-computer interaction technology and practices. The primary purpose of the review is to determine whether research and practice in this area have evolved sufficiently to support the development of a human factors guide to computer system design. It is concluded that insufficient data exist for the development of a "quantitative reference handbook" in this area, but that a "human factors design guide" -- which discusses issues, alternatives, and methods in the context of the design process -- is both feasible and needed.

UNCLASSIFIED
SECURITY CLASSIFICATION OF THIS PAGE(When Data Entered)

TABLE OF CONTENTS

TABLE OF CONTENTS (continued)

TABLE OF CONTENTS (continued)

LIST OF FIGURES

INTRODUCTION

This report is one of the products of a rather broad survey of the literature on human factors in computer systems. The survey was conducted under research contract N00014-76-C-0866, funded by the Engineering Psychology Programs Division of the Office of Naval Research. The principal purpose of the survey was to assess the current state of our knowledge in this area, in order to determine whether that knowledge is adequate to support the development of a human factors guide to the design of interactive computer systems. Although this report provides, to some extent, a descriptive and critical outline of the literature surveyed, its principal focus is on the issue of design guidelines.

THE PROBLEM

Human factors in computer systems is by nature a highly interdisciplinary field. As a result, the relevant literature is widely scattered. Those who would perform effective application work in this area must be aware of a broad range of literature, including: (a) the general literature on human factors, as well as that concerned specifically with computer systems, displays, data entry, and specific application areas; (b) a significant segment of the basic psychological literature, including especially the areas of perception, information processing, and cognitive psychology; and (c) a significant segment of the computer science literature, including especially the areas of batch and interactive languages, display and input devices and techniques, and specific application areas.

If the maintenance of such a broad familiarity with the literature is difficult for the specialist in this area, it is all but impossible for the typical interactive system designer. The designer is usually a computer scientist or, less frequently, a specialist in the application area to which computers are being applied. Although the designer may recognize that many design decisions have human factors overtones, he or she has typically had little or no exposure to the psychological or human factors research literature.

In some well established research areas, such as keyboard design and certain physical properties of displays, guidelines exist which are reasonably good and fairly detailed. Such guidelines may be quite helpful in the design of a console or other interface device for a system, or even in the selection of an appropriate off-the-shelf input/output device. As we progress toward the more central issues in interactive systems, such as their basic informational properties, user aids, and dialogue methods, available guidelines become sketchy and eventually nonexistent. The interactive system designer is given little human factors guidance with respect to the most basic design decisions. In fact, the areas in which existing guidelines concentrate are often not even under the control of the designer, who may have more freedom with respect to dialogue and problem-solving aids than with respect to terminal design or selection.

Between these two extremes lies a middle ground in which a few attempts have been made to develop guidelines. Engel and Granda (1975) have produced an excellent, but very brief, set of guidelines which addresses some of the dialogue issues and touches on display formatting. The International Purdue Workshop on Industrial Computer Systems (1975) has produced a document intended for use by designers of process control systems. This document is oriented around the design process, and has several desirable features, but it is unfortunately weak in the human factors area. Several other authors have written less formal discussions which deal with desirable dialogue properties and related matters. No guidelines have been found, however, which are both extensive and sound.

The dearth of human factors guidelines with respect to interactive problem-solving methods, user aids, etc., comes as no surprise. The concentration of existing guidelines on person-computer interface hardware and low-level dialogue issues clearly reflects the historical emphasis of the field. Only fairly recently has a really strong emphasis on cognitive issues become apparent in the practice of human factors. The shift in emphasis results in part from the increasing use of interactive systems to

support cognitive tasks, and in part from the emergence of cognitive psychology and human information processing as stronger areas of basic behavioral theory and experimentation. Undoubtedly, it also reflects the increasing maturity of human factors as a professional discipline.

More extensive human factors guidance for interactive system design is sorely needed. The computer science community is becoming increasingly aware of the significance of human factors issues in interactive systems, but the available options are limited. Newman (1977) is undoubtedly correct in asserting that psychologists and system designers complement one another, and that the most effective approach is a direct involvement of human factors personnel in the design effort. Yet very few psychologists are sufficiently well-grounded in computer science to achieve acceptance in this role, and the number of new interactive systems being developed is staggering. Pew and Rollins (1976) have tried an approach in which human factors personnel define a dialogue-specification technique, building appropriate human factors constraints into a specification method which is then used by the system designers. This approach may allow a limited number of human factors personnel to exert a larger design influence than is possible by direct involvement in the design. It is not clear, though, that this approach is applicable in really complex systems, or cost-effective in small systems. In any case, this approach concentrates on the form of the dialogue, and does not address more basic issues of user information requirements, selection and design of appropriate problem-solving aids, etc.

The development of human factors guidelines for use by system designers appears to be the only viable solution of this problem. As has already been noted, though, existing guidelines are somewhat primitive. The principal goal of this study was to determine whether or not the state of the art in human factors in computer systems is currently adequate to support the development of more extensive guidelines. A secondary goal was to determine what form such guidelines should take, if indeed their development is feasible.

THE LITERATURE SURVEY

In order to investigate these questions, an extensive literature survey was undertaken. A companion report (Ramsey, Atwood, and Kirshbaum, 1978) describes the survey in detail, including the areas covered and those excluded, as well as the survey method employed. That report, which is a bibliography of the most relevant literature, contains descriptions and critical commentaries on 564 papers.

Briefly, the literature surveyed was concerned with such areas as interactive input and output devices and techniques (and, to a limited extent, batch input/output devices and techniques, as well); people-computer dialogue (including languages and dialogue methods); analysis of user and task properties; techniques for user requirements analysis, system design, and people-computer system modeling and evaluation; and people-computer problem solving. In an effort to keep the survey manageable, some areas were generally excluded from consideration even though they may intersect the broad area of human factors in computer systems. These topics include personnel selection, training, and management; documentation; social and organizational implications of computers; software development; and literature describing specific systems.

An important aspect of the survey was its attempt to include relevant articles not only from the human factors literature, but also from the basic literature of both behavioral science and computer science. If the ultimate goal of this effort is the development of guidelines useable by interactive system designers, it is clear that one must consider not only knowledge gained through behavioral experimentation, but also current and projected practices in people-computer interaction.

The literature survey was quite extensive. Some 20,000 citations were considered, of which more than 4,000 were found to be at least loosely relevant. The original intent of the survey was to be quite exhaustive.

However, resource limitations, as well as difficulties in obtaining copies of a few papers, have precluded detailed consideration of some papers which are probably directly relevant. When resource limitations have been an issue, breadth of coverage has been the primary criterion for selection of the 564 papers actually considered in detail. For the purposes of this study, acquisition of a sound overview of the literature was considered more important than a comprehensive treatment of any particular specialty area within the field.

GUIDELINES AND THE DESIGN PROCESS

When the phrase "interactive system design" is used in this paper, the reference is to the process whereby the functional behavior of an interactive system is determined. The decisions made in this design process involve questions of basic system informational properties from the user's viewpoint, and of interactive dialogue, display formatting, etc. The design activity with which this paper is concerned is often called "requirements analysis" or "functional system design." This study is not concerned with the later "software design" process in which the design decisions involve software modules, files, etc.

The emphasis, among developers of human factors guidelines, has been placed heavily on quantitative reference handbooks. That emphasis is consistent with a desire for high quantitative precision and with the expressed preferences of designers -- though not necessarily interactive computer system designers -- for concise, preferably graphical, reference data (e.g., Meister and Farr, 1967). For such handbooks to be satisfactory, however, at least two conditions must be satisfied.

First, the handbook must be based on an extensive collection of raw quantitative data pertaining to operator performance and operator-system interaction. This requirement has led to widespread advocacy of large-scale parametric experimentation (e.g., Granda, 1977; Meister, 1977) for the purpose

of constructing such a data base. Such an undertaking is expensive, but sorely needed, in the interactive computer systems area, as well as in many other person-system domains. One clear conclusion of this survey effort is that quantitative data on user performance and user-system interaction are few and fragmented. A quantitative human factors reference handbook for interactive systems design appears to be well beyond our current capability.

There is a second assumption underlying the desire for quantitative reference handbooks, however. The design decisions for which they are used must be explicitly identified by the designer. In "classical" system design situations, this may be a reasonable expectation. When a design engineer encounters issues such as level of illumination, size of control knobs, or type and position of a numerical display, the design decision involved tends to be: (1) quantitative, (2) explicit, and (3) clearly associated with the "operator interface", and therefore closely identifiable with human factors. Even here, there is evidence which suggests that designers may proceed without consulting readily available human factors handbooks (Meister and Farr, 1967).

In the design of interactive computer systems, virtually every decision which affects the functional behavior of the system has direct human factors overtones. This claim can be made in the cases of automobiles, aircraft, and radios, too, but only in a much weaker sense. In a system whose basic function is communication with a user, and whose basic purpose often is to assist the user with tasks which are cognitive or informational in nature, human factors issues pervade the entire design process.

Furthermore, the important design decisions may not be explicitly recognized. It appears that the primary problem-solving method employed in design tasks is an "analytic/synthetic" approach. This approach involves an analysis of the design problem, in which the components of the problem are identified, followed by the synthesis of a solution. It does not

involve an explicit search for alternative solutions; instead, a solution is "synthesized" based on pattern recognition and the use of components from past solutions to other problems. While this characterization of designer behavior is somewhat speculative, it appears quite compatible with a variety of observations of an empirical (e.g., Meister and Farr, 1967), theoretical (e.g., Simon, 1969), or pragmatic (e.g., Granda, 1976) nature.

If, in interactive systems, the design decisions which involve human factors issues are pervasive, implicit, and often qualitative, a reference handbook is unlikely to be effective. However, there is another mechanism for providing human factors guidance to the system designer, which appears to be more compatible with the design process and the designer's information needs. It also appears to be more compatible with the current state of our knowledge. This approach is the design guide.

As the term will be used here, a "design guide" is a document whose *structure parallels the major* steps of the design process itself. Although such a guide should be indexed sufficiently that it can be consulted with a specific question after that question has been identified, that is not the design guide's primary mode of use. It is intended to be consulted during the analytic process, before design decisions are made. Such a guide must identify issues, suggest alternatives, and present (where they exist) hard human-factors data at the point in the design process at which this information is most relevant. To make the concept somewhat more concrete, Figure 1 suggests some possible major sections which might be contained in such a document.

Clearly, this is more of a "how-to-do-it" document than has typically been undertaken by human factors personnel. Such an approach appears justified, however, by several factors: (1) the pervasiveness of human factors issues throughout the interactive system design process; (2) the likelihood that a reference handbook would not be consulted and used effectively; (3) greater compatibility with the problem-solving behavior and information needs of the system designer; (4) inadequacy of the existing human factors data base to support development of a quantitative reference handbook; (5) comprehensiveness,

Users: their behavior in general; how to determine the properties of a particular user population; the implications of those properties for the interactive system.

Tasks: what tasks users perform; how to determine tasks involved in an application.

Requirements analysis: How to analyze information requirements; how to select appropriate types of problem-solving, clerical and user support aids; allocation of basic tasks to user or computer; modeling of user-system interactions; evaluation of basic design.

Interactive dialogue: properties of different dialogue types; selection of appropriate dialogue type(s); detailed design of command language, system access structures, tutorial aids, etc.

Output devices and techniques: properties of display devices; implications of dialogue method for display device selection; selection or design of display device(s); detailed display design, formatting, coding techniques, etc.

Input devices and techniques: properties of input devices; implications of dialogue methods for input device selection; selection or design of input device(s).

Evaluation of system performance: use of subjective evaluations, objective performance measures.

Figure 1. Possible Section Topics for a Design Guide

in the sense that this format allows discussion and some guidance even in areas in which specific experimental data are lacking; and (6) the fact that this general format is capable of encompassing not only human factors data, but also applied human factors methods.

Ordinarily, conclusions are presented in the "Conclusions" section of a report. In this case, however, it seems necessary to forewarn the reader. The following sections discuss the surveyed literature from the point of view of its applicability to design guidelines. The literature must be considered with both types of human factors guidance in mind. Either type might have been supportable by the literature, or both, or neither.

USER AND TASK PROPERTIES

USER NEEDS, CAPABILITIES, AND LIMITATIONS

As Meister (1976) points out quite lucidly, an interactive system acquires its characteristics from the goals adopted by its designer. If the system is to be effective, the system goals must be compatible with the needs and capabilities of the individual user. While this compatibility may be achievable, at least in part, through operator selection and/or training, it is its achievement through appropriate system design that is of concern here.

If compatibility between the system's purpose and the user's needs and capabilities is critical to system success, the designer must somehow be made aware of those needs and capabilities. This does not appear to have been accomplished in any effective or systematic way. The literature most commonly used by interactive system designers contains many such prescriptions as, "Make the system easy to use," but contains few explicit treatments of user capabilities, limitations, or habits, either in general or by specific user population. The systems design literature does contain discussions of user requirements analysis methods (see next section), especially interview and simulation methods, but the use of these techniques does not obviate a need for basic data concerning user behavior.

Clearly, the behavioral issues involved here are complex, subtle, and, in many respects, poorly understood. At the most basic level (from the viewpoint of the designer), user acceptance, user confidence, and good user-system performance are all desirable or even necessary, but may, under some circumstances, be conflicting goals. For example, Dickson et al (1977) compared raw-data displays with statistically summarized displays in a management information system. Although better decisions resulted with summary displays, the decisions took longer, and user confidence in the decisions was lower. Artifical lockout, in which the user is required to wait (several minutes) between requests for computation, has been found to improve performance in a problem-solving

task (Boehm et al, 1971; Seven et al, 1971), but was also found to cause considerable user dissatisfaction. Franklin and Dean (1974) describe user reactions to a graphical system for electronic circuit design. The intended users (students) reportedly found the system easy to use, but avoided using it because they were intimidated by its rapid response and the belief that they were tying up expensive computational resources.

In the face of extremely complex behavioral issues, ready-made prescriptions often cause more difficulties than they correct. It would appear to be necessary to convey to the designer a better understanding of the behavioral issues themselves, and of the variables which most directly influence user behavior. Carried to an extreme, this implies a curriculum in psychology. There does appear to be a middle ground, however, which is tenable and potentially effective.

Strengths and Weaknesses of Humans and Computers

A simple (and often simplistic) approach to this problem is the listing of strengths and weaknesses of both humans and computers. Such lists are common, and often contain such prescriptions as "humans are good at decision making; computers are good at computation." Aside from being so vague as to contain little useful information, these comparisons may even be misleading (see Vaughn & Mavor, 1972).

A related, but much more satisfactory approach is represented by Miller (1967). Miller describes the problem-solving and information-processing capabilities of the human user of a computer system in a very readable way. The characterization is reasonably good, but is subject to the criticism that it is not sufficiently specific to be of clear relevance to the designer. The only obvious way of making such discussions more specific is to limit the discussion to specific tasks which might be performed by the user. Crawford et al (1977) take a worthwhile step in this direction. Real progress in this direction will require a more effective taxonomy of operator tasks than presently exists. As a later section will show, this is currently a very weak area, in spite of several attempts to develop such taxonomies. In the foreseeable future, the best approach may be a description of operator capabilities and

limitations with respect to a subset of fairly explicit tasks associated with a particular class of interactive system (e.g., message handling, bibliographic search).

Behavior of Specific User Classes

Investigations of the behavior and needs of specific user classes are rather primitive, at present, but nonetheless have produced results which are of significance to the designer. The most informative investigations of this sort have been the reports of the "MICA survey" (Eason, 1976; Eason et al, 1975; Stewart, 1974, 1976b). Three major user classes have been subject to considerable discussion in the literature: (1) "naive" (or computer-inexperienced) users, (2) managers, and (3) scientific and technical users. Brief discussions of some major behavioral properties of these user classes are shown in Figure 2.* In addition to these user classes, there are scattered discussions in the literature of consumers (viewed as indirect system users; Ivergard, 1976), non-English-speaking users (Hanes, 1975), handicapped users (Raitzer et al, 1976), and even illiterate users (Evans, 1976).

Naive Users

The naive user has been subject to considerable discussion in the recent literature. The interactive system designer, who is necessarily rather sophisticated with respect to computers, often has great difficulty producing a basic system design, and particularly a detailed dialogue design, which is comprehensible and easily useable by the relatively naive user. Aside from a number of special abilities which programmers probably have as a result of self-selection, they also have a great deal of specialized knowledge of system behavior which affects the way in which they interact with systems. For example, an understanding of the interfaces among operating system and application modules often

* In this and subsequent tables, a single asterisk (*) indicates a reference which presents survey or questionnaire data or summarizes experimental results concerning the topic of the table. A double asterisk (**) indicates a reference which presents user performance data or relatively detailed results of controlled experimental study(s) on the topic.

User Class	Discussion	Principal References
Naive users (inexperienced with computers)	Computer-naive users are actually a very heterogeneous group, but have many common properties. Naive users benefit greatly from computer-initiated dialogue, usually require more tutorial features. Correct implicit "mental model" of computer systems and interactive dialogue cannot be assumed, must be explicitly conveyed by system. Naive user population has many detailed implications for dialogue design. Smooth transition from naive to experienced user is often difficult in current systems.	Card et al (1974)** Eason et al (1975)* Evans (1976) Martin (1973) Nickerson & Pew (1971) Thompson (1969, 1971)
Managers (including military commanders, etc.)	Managers tend to have highly variable information needs; current systems are often too rigidly constraining to satisfy those needs. Managers tend to place high negative value on own effort, *have considerable discretion with* respect to mode of system use or nonuse. Thus, very low "impedance" is required to capture manager as direct user. If dissatisfied, manager tends to resort to "distant use" (interposing operator between manager and system) or partial use.	Ackoff (1967) Alter (1977)* Eason (1976)* Igersheim (1976) Raichelson (1972)
Scientific and Technical	High proportion report dissatisfaction with available automated tools. These users often respond to such dissatisfaction by becoming personally involved in design or implementation of software tools, or by altering task to match available tools.	Stewart (1974, 1976b)*

* Presents survey or questionnaire data or summarizes experimental results.
** Presents user performance data or results of controlled experimental study.

Figure 2. Properties of Some Major Classes of Interactive System User

allows sophisticated users to correctly interpret otherwise ambiguous error messages on the basis of context cues not useable by the naive user.

While the design community is aware of this problem in general, discussions of appropriate design properties are often quite vague ("make the system easy to use"). With appropriate caveats, much more specific guidance is possible. For example, the use of computer-initiated dialogue (in which the computer presents alternatives, questions, or forms, and the user responds) is almost always preferable for naive users, unless fairly sophisticated knowledge-based or natural-language techniques are used (see Thompson, 1969, 1971; and Martin, 1973 for discussion). Such practices as unnecessary abbreviation, and command languages with positional parameters, probably have more significant negative effects with naive users than with experienced users. Engel and Granda (1975) and Nickerson and Pew (1971) offer a number of guidelines with respect to command language design, user feedback, error message contents, system behavior in the event of errors, etc., which are especially applicable to naive users. Appropriate use of embedded training (see later section) and other tutorial features distributed throughout the dialogue can be of assistance with naive users.

If the naive user is a one-time user, or occasional user, of the system, the use of a simple, clear, computer-initiated dialogue may be sufficient to ensure satisfactory results (see, for example, Evans, 1976). For the repeating user, however, dialogue simplicity may be a two-edged sword. In one of the few empirical studies of naive users, Eason et al (1975) report that overemphasis on dialogue simplicity often prevents the user from utilizing the full computational power of the system. As the user acquires experience with the system, the constraints imposed by the dialogue become burdensome. A frequent practice in modern systems is the provision of two dialogue modes: (1) a simple, highly tutorial, computer-initiated mode for the beginning or naive user, and (2) a highly abbreviated, user-initiated mode for the experienced or computer-sophisticated user. Only rarely is any mechanism provided for achieving a smooth transition between these two modes as

the user acquires more experience. The findings of Eason et al would suggest that this may be a significant failing.

Sophisticated users, including system designers, are often unaware of the specialized knowledge, context cues, etc., which they use in interacting with computer systems. It is conceivable that the designer might benefit from an explicit discussion of the differences between naive and sophisticated users. Ideally, such a discussion should be based on empirical analyses of information utilization by both classes of user. No such analyses, and no such comparative discussion, were found in the survey.

Managers

The term "manager" is used broadly here, and includes not only business executives, but also military commanders and others in an organizational management role. Managers are usually naive users, in the sense discussed above, but possess enough additional common characteristics to warrant consideration as a separate class. In fact, the specialized needs of these users has spawned a large literature describing "lessons learned" and systems analysis techniques for management information systems (MIS's). Once again, though, little has been done in the way of formal empirical study.

In interacting with a computer system, the user incurs certain costs, not only in dollars, but also in terms of personal time and effort. In return, the user receives certain benefits, which may include improved solutions, savings of time and effort relative to other solution techniques, etc. Computer users in general appear to behave as if they perform implicit cost-benefit evaluations based on their own perceptions of costs and utilities (see Carbonell, 1967, for discussion). Managers are no exception, but they appear to place a particularly high negative value on personal time and effort, and when compared to other user classes, have more "discretion", or authority to reject the system or use it differently than intended (Eason, 1976). These two factors make it particularly difficult to "capture" the manager as a direct user of an interactive system.

There is also some evidence that the information requirements of managers are more variable than those of most user classes. Eason (1976) has noted this problem in connection with business MIS's, and Raichelson (1972) reports a similar problem with command and control systems. Both suggest that greater system flexibility is required, but differ in the means suggested for achieving it. It is especially interesting to note that past MIS failures have led to widespread advocacy of more -- and more formal -- system analysis techniques aimed at extracting specific information requirements from managers prior to MIS design. While these methods may be very helpful in providing the designer with a general understanding of the environment, Eason's survey results and Raichelson's observations would suggest that this approach cannot, in principle, solve the basic problem. A later section on system flexibility will address this issue further.

Scientific and Technical Users

Scientists, engineers, etc., constitute a third user class which has received some specialized attention. Allen (1969) and Back (1972) discuss the information requirements of such users from the point of view of general scientific communication, but do not relate those needs to the use of interactive tools. On the basis of the MICA survey, Stewart (1974) indicates that most specialist users report difficulty using on-line systems due to inadequate workstations, terminal operation problems, or difficulties with software and operating procedures. In another paper, Stewart (1976) characterizes these users as utilizing computerized tools in three broad ways: (1) use of predefined analytical packages, (2) definition of specialized programs to be implemented by others, and (3) hands-on programming. In fact, it may be the response of these users to inappropriate or inadequate automated tools which most strongly differentiates them from other classes.

User Responses to Inadequate Systems

Another result of the study of specialized user classes has been

the development of a gross characterization of user responses to inappropriate or inadequate systems. A list of common responses, due primarily to Eason (1976) and Stewart (1976) is presented in Figure 3. As indicated in the figure, user responses vary with the role, status, and knowledge of the user. For example, a very common response of managers to systems which fail to totally satisfy their needs, or are simply difficult for the managers to operate, is the interposition of another individual between the manager and the system. That individual is responsible for direct operation of the system and satisfaction of the manager's information needs. Eason contends that this is an unsatisfactory response, as it increases the effective response time of the system, and the role of intermediary is very difficult to fill. Alter (1977), on the other hand, believes that the use of an expert intermediary may be a desirable approach. In any event, the designer must be aware of the possibility of such "distant use". If the system is designed for the manager as direct user, but is in fact operated by an expert intermediary, the dialogue method, and perhaps other system properties, are almost certain to be inappropriate.

Scientific and engineering users tend to respond to inadequate tools in an entirely different way, becoming progressively more involved in the design, and even programming, of the systems they use. Having observed computer-aided problem-solving practices in this user community, Stewart (1976) suggests that this is an undesirable situation. Obviously, this can be a costly practice, but Stewart's concern is more subtle. He suggests that those who become heavily involved in tool development may apply their tools to new problems whether they are appropriate or not. The resulting psychological set may interfere with problem solving, and may even influence problem selection.

Statistics on Use of Time-Sharing Systems

There exist a large number of statistical studies of user behavior when interacting with specific interactive systems. Typically,

Response	Definition	Comments
Dis-use	Reliance on other information sources.	Requires existence of alternative information source, user with sufficient discretion.
Mis-use	"Bending the rules" to shortcut operational difficulties.	Requires significant knowledge of system. May negatively impact system integrity.
Partial use	Use of (perhaps inappropriate) subset of system capabilities.	Users frequently adopt "satisficing" strategy, may not learn most relevant system capabilities.
Distant use	Interposition of operator between user and system.	Requires high status and discretion. Typical response of managers.
Modification of task	Changing the task to match capabilities of system.	Prevalent when tools are rigid, problem is unstructured, as in scientific problem solving.
Compensatory user activity	Compensation for system inadequacies by additional user actions.	Typical with users of low discretion, as clerks.
Direct programming	Programming by user, in order to modify system capabilities to suit needs.	Typical response of computer-sophisticated user, as scientific and engineering user.
Frustration and apathy	Response of user when above actions are inadequate or unsatisfactory.	Involves lack of user acceptance, high error rates, poor performance.

Figure 3. User Responses to Inadequate System

these data have been gathered automatically by monitors built into a time-sharing operating system. Figure 4 describes a representative sampling of these studies. In general, the data reported consist of descriptive statistics concerning the temporal properties of user sessions, and frequency distributions for usage of particular system processors or commands. There is good reason to believe that these data are not very generalizable from one system to another, because of very large variations in system and user population properties. These statistical summaries are ordinarily obtained for use in response time and system utilization planning studies, and they are typically published as an internal report by the institution involved. It is conceivable that a serious attempt to collect and integrate these reports would yield useful information. With a few exceptions (e.g., Boies & Spiegel, 1973; Carlisle, 1972), the discussions are purely descriptive, and provide little real behavioral insight.

User Errors and Error Analysis

Errors constitute an aspect of behavior which is of particular importance to the designer. Very large reductions in error frequency and severity can be achieved if the system design is based on an adequate understanding of the types and causes of errors which occur in the use of the system, or which occur elsewhere in the process, of which the system is a part. A poorly designed system, on the other hand, may not only fail to provide assistance to the user in particularly error-prone situations, but may actively induce user errors.

The standard measures used in human performance studies are speed and accuracy, and almost all such studies contain information on errors. These data are often highly task-specific, however, and the integration of such information is no easy task. Most of the explicit discussions of user errors found in the survey deal with data entry. Relatively few discussions were found of errors in computer-aided problem solving, or even errors related to interactive dialogue, and no integrative summary was found in these areas.

System	Data Reported	Reference
JOSS	Statistics on input/output quantities and rates, user session temporal properties, space/time properties of executed programs.	Bryan (1967)**
TSS/360 Program Control System at IBM research center	Statistics on frequencies of use of commands by class (e.g., data management, editing, debugging) and specific command. Discussion of user competence groups.	Boies & Spiegel (1973)**
IBM/360 Time Sharing System	Statistics on duration and frequency of terminal sessions, use of language processors, command usage, and user response time.	Boies (1974)**
IBM/360 Batch Processing System	Statistics on utilization of compilers and system resources, temporal distribution of jobs and CPU utilization by day, week, and year	Haralambopoulos & Nagy (1977)**
AUTOGRP medical statistical analysis system	Frequencies of use of commands overall and by user. Comparison of command usage for beginners and experienced users.	Carlisle (1972)**
TENEX time-sharing system	Statistics on session duration, CPU utilization	Grignetti et al (1971)**
MULTICS system	Frequencies of use of system commands, with statistics on associated "think times." Distributions of think time.	Rodriguez (1977)**

Figure 4. Representative Examples of Statistical Studies of User Behavior in Time-Sharing Systems

A few scattered papers deal with error-prone properties of query languages and query problems. For example, some constructs which appear to be error-prone are disjunction and negation (L. A. Miller, 1974), re-expression of inequalities (Gould & Ascher, 1975), and logical set relations (Thomas, 1976). A larger -- though not necessarily more useful -- literature exists in the area of computer programming errors and error-prone programming-language properties, but that literature was not within the domain of the survey.

The fact that data entry dominates the discussion of user errors is not necessarily discouraging. Data entry is a major source of costly errors, and this literature contains several good models for the presentation of error data. There appear to be at least four forms in which information about user errors might usefully be conveyed to the designer. The first is a simple tabulation of error rates for particular tasks. With few exceptions -- possibly Hinds (1960) error-rate data for punched-card data entry, for example -- the tasks for which such data are available appear to be too specialized, or too vague, to provide useful information for the designer in raw-data form. This probably results in part from the absence of an adequate taxonomy of user tasks. If such a taxonomy existed, error data (and other performance data) expressed in terms of such tasks might be directly applicable.

There are two very good -- and very different -- summaries which deal with data entry errors at a level which might be useful to the designer. Seibel's (1972) review is not explicitly about errors, but discusses alternative data entry devices and procedures in terms of their effect on speed and errors, and is probably the best existing integration of the information available on alternative encoding techniques, transcription, typing, etc.

Bailey's tutorial presentation (Bell Telephone Laboratories, 1973) is an attempt to educate the system designer with respect to the nature and causes of user errors in data entry systems, with an

emphasis on the corrective measures available to the designer. The paper is highly readable, but this approach is necessarily rather verbose. A tutorial presentation of this sort is probably capable of exerting more influence on the system designer who has no human factors experience than is either of the previously discussed presentations.

Given that error data do not yet exist for most user tasks in a form useable by the designer, procedures for error analysis during system design may be the most useful information which can be provided. Nawrocki: et al (1973) describe a "qualitative" approach to error analysis. A procedure is described for categorizing errors, relating them to the components and processes in which they occur, and determining their causes. The case studies by which the technique is illustrated all involve data entry, but the technique appears more broadly applicable. Bailey (1972) describes a procedure in which a very detailed simulation of a system is conducted, after design and preparation of complete user documentation, training materials, etc., but before implementation. The emphasis of the simulation is on analysis of error rates and causes, and the technique is undoubtedly effective for this purpose. The expense, and particularly the increased span time required by this approach, may be considered prohibitive by many.

Clearly, conceptual errors by the user are much more difficult to deal with than are simple errors of encoding, transcription, etc. If the designer is aware of the nature of errors of the latter type, it is often possible to detect, and perhaps even to correct, the errors by automated means (see Obermayer, 1977 and Wasserman, 1973 for two different applications of automated error detection, reporting, and correction). Even in the case of data entry, though, many of the errors are conceptual, involving, for example, erroneous descriptions of events. Strub (1975) found that over 80% of the errors observed in a tactical data entry task were not detectable by existing automated techniques. Artificial-intelligence techniques may eventually be very helpful here, but are presently useful only in highly restricted task domains.

Individual Differences

Individual differences among users, involving such variables as abilities, acquired skills, general background, sex, and age, may affect performance in tasks involving interactive systems, and may, therefore, have implications for system design. Various researchers have reported that performance correlates with age, attitude measures, and mechanical aptitude (Weltman et al, 1973), sex and typing skill (Earl & Goff, 1965), vocabulary test performance (Morrill et al, 1968), recency of training and class standing (Robins et al, 1974), and a wide variety of other variables. Others have speculated that performance relates to cognitive decision style (Strub & Levit, 1974; Levit et al, 1975) and general intelligence (Carlisle, 1974), but have not observed such performance differences in subsequent empirical studies. Responses to certain system properties ("interface complexity", Carlisle, 1974; "interface flexibility", Walther, 1973) have been found to vary with user properties.

The principal difficulty with making use of such data is that the observed effects are often highly task-specific. Most of the reports of individual difference effects are the results of investigations of the properties or performance of particular systems. Most of these studies have not been intended as explicit investigations of individual differences, and few, if any, of them have involved any systematic variation of the task performed. Yet any attempt to integrate their results leads one immediately to the conclusion that the relevant individual difference variables are strongly dependent on the particular task performed. Sometimes, the individual difference variables are even further confounded, as when female clerk-typists are found to perform better than male engineers on a data entry task involving typing.

Theologus et al (1970) have studied the specific ability requirements of a number of jobs (not specific tasks). The results may be useful for personnel selection, but are not sufficiently detailed, in terms of tasks, to be useful to the designer, nor are they specific to computer-system related jobs. This study may, however, provide some insight into task-ability relationships.

Given the sparcity of relevant data relating individual differences to system design considerations, it would appear that nothing quantitatively definitive can currently be written for the designer on this topic. However, Wylie et al (1975) have provided a good tutorial discussion on the causes and effects of individual differences, particularly as they affect the design process. This discussion is written so as to be useful for the designer, and may be the best we can do at present.

TASK PROPERTIES

There are a very large number of tasks which may be performed by the users of interactive computer systems. Some of these tasks would exist independently of the computer system, and are supported in some way by the system, while others (logging in, for example) are by-products of the computer system itself. Human performance varies strongly with task properties. Appropriate automated aids, dialogue methods, etc., clearly vary with the task. Research results on human behavior often fail to generalize to other tasks, even when those tasks are superficially similar to the experimental task. While attempts have been made to develop task taxonomies, few existing taxonomies address user tasks at a level which is useful in the functional design of interactive systems.

Bennett (1971) suggests that a "task taxonomy requires at least the following levels of units: a 'job', perhaps definable as a full-time paid position; a 'function,' a subdivision of a job, or a grouping of tasks (such as the research portion of a professor's job); a 'task', an undefined but fairly specific set of activities (for example, 'calibrate the equipment'), and a 'subtask' a subdivision of a task (e.g., 'adjust the knob')". Using this terminology, most of the work which has been done has been at the "function" or even "job" level, while meaningful discussions of human performance must take place at the "subtask" or "task" level.

Consider "reading", for example. For many purposes, this would be considered a very low-level user task. And yet the user may be reading continuous text from the display for comprehension, or reading continuous text to detect errors, or searching a menu for the desired command, to cite a few alternatives. There is evidence that these specific tasks may vary in their requirements for minimum display resolution, line spacing, use of upper and lower case characters, and perhaps other variables.

Obviously, the development of a taxonomy of user tasks down to this level of detail would be quite difficult. It may only be practical to attempt such taxonomization within sharply delimited application areas, such as text processing, message handling, or surveillance. However, if guidelines are to exist which provide quantitative data concerning human performance, and if such data are to be applied by the designer, some common understanding of user tasks is absolutely necessary.

Of the papers surveyed, several contained explicit attempts to taxonomize user tasks at one level or another. At the global end, Theologus et al (1970) analyzed jobs in terms of their requirement for specific known human abilities. This paper provides some useful insights, but is strictly at the "job" level (e.g., computer programmer). R. B. Miller (1965, 1969) discusses very general problem-solving tasks, along with the relationship between tasks and information requirements. This task characterization is not specific enough for the purposes outlined here, but the later paper, in particular, contains a good discussion of the relevance of a task taxonomy to system design. Nicholson et al (1972) attempt to taxonomize user tasks in Navy interactive computer systems. Their taxonomy contains general interaction operations, at a level which is somewhere between "task" and "function", and refers to some problem-specific tasks, as well. This would appear to be a good way to start, if the goal is a general user task taxonomy. Unfortunately, the result doesn't quite reach a level of detail adequate for meaningful discussions of human performance.

Wylie et al (1975) consider only functions performed in connection with automated surveillance systems. The result is a taxonomy which appears to be very good, and achieves sufficient detail to support a discussion of human performance and automated aids. While an even more detailed task breakdown would be desirable, this study appears to be a notable success in an area which has seen few really useful products. It is important to note that the tasks discussed here (e.g., "alerted searching for undetected lower order features") are very

specific to surveillance, and have little in common with the same-level tasks involved in, for example, text processing. Restricting the scope of inquiry to a single application area may be the key to success in task taxonomization.

There is a second advantage to the development of separate task taxonomies for specific application areas. If tasks are identified in a very general way, the burden of relating them to the requirements of a specific system falls -- perhaps somewhat heavily -- on the designer. If the tasks are characterized in terms of a particular application area or process, the designer may find it much easier to recognize and relate to them. Thus, the taxonomy may be more easily used, and a need for formal task analyses, by the designer, may even be eliminated.

In summary, our lack of success -- some might say lack of effort -- in characterizing user tasks is probably the greatest existing deficiency of the field. There is some indication that this is a manageable problem, however, if approached with a limited application area in mind. In any case, progress in this area is necessary if we are to provide detailed, quantitative guidelines which allow the designer to confidently apply the results of human factors research to the identification of appropriate automated aids and their detailed specification.

DETERMINATION OF USER REQUIREMENTS

REQUIREMENTS ANALYSIS METHODS

Specific research results, comparing input devices or display features, for example, represent an important kind of contribution which human factors practitioners can make to the system design process. Before such data can be applied, however, it is necessary that the designer have a basic conception of the functional requirements of the system. The previous section has indicated that some of these requirements may be suggested by an isolated consideration of the properties of the task, or of the user. All too often, user requirements analysis is no more detailed or formal than this. Commonly, the designer uncritically accepts the user's stated preferences, often obtained via interview, integrates that information with the designer's own prior conception of the user's task and work habits, and proceeds to design the system. This is a particularly prevalent scenario in those cases in which the designers are themselves expert in the application domain of the system. This approach <u>can</u> produce good systems. Its success is limited, however, by the validity and completeness of the designer's conception of the task, the validity and completeness of the user's view of the task, the ability of users to verbalize their knowledge, and a wide variety of subjective biases which may influence the behavior of the user or the designer.

It would be presumptuous to suggest that there are any guaranteed solutions to this problem. On the other hand, the observation, description, modeling, and improvement of human performance is what behavioral research is all about. In a loose sense, user requirements analysis is itself a behavioral research problem. Many of the observation and analysis techniques developed by behavioral scientists for use in basic research have been effectively adapted and applied by human factors personnel to the analysis of user requirements for person-machine systems. The computer systems design literature contains scattered references to these methods, but suggests that few designers have any broad awareness of these techniques, or of their strengths and pitfalls. It appears

appropriate, then, for a human factors design guide to provide methodological assistance to the designer.

The use of appropriate requirements analysis techniques can not only result in more accurate identification of user requirements, but can also serve to increase the involvement of the user in the design process. While this can improve the design itself (Eason, 1976, 1977; Kriebel, 1970; cf. Greenwald-Katz, 1976), there is also evidence that it results in greater user acceptance of the resulting system (Eason, 1977; Igersheim, 1976). Eason (1977) suggests that active involvement of users in the design process can also result in delays, and difficulty in translating stated user wants into precise specifications, which can in turn be somewhat frustrating to the designer. It is not clear, however, that user involvement in the design itself is necessary to achieve the beneficial effects. It seems likely that greater designer-user interaction in the requirements data collection phase is a desirable compromise.

Data Collection

Data collection is the first step of the user requirements analysis process. Basic approaches to data collection include interviews, questionnaires and surveys, field observations, simulation, and gaming. Figures 5, 6, and 7 present a general overview of some of these techniques, with comments and references. In most cases, the references involve applications of the techniques in the specific area of interactive computer systems design. In a few cases, however, the reference is to a general discussion of the method, since no detailed discussion of the technique in connection with interactive computer systems was found. These instances are identified in the "comments" section of the figures.

The computer science literature contains a large number of papers dealing with requirements analysis. In general, those papers emphasize the kinds of information the designer should look for (e.g., mission, environment, organizational structure, decision maker style),

but often at a relatively high level (see, for example, the review of management information requirements analysis techniques by Taggart and Tharp, 1977; see also Evans, 1967). This literature provides relatively little discussion about how the designer should obtain this information. Most of the emphasis is on interviews and questionnaires, with limited discussion of more objective, empirical, or performance-based techniques (for a basic discussion of interview and questionnaire techniques, see Parker, 1970).

Some specific types of questionnaire and survey methods are discussed in Figure 5. While these methods can be useful for determining user information requirements, their uncritical use is potentially hazardous. The principal difficulty is the subjective nature of questionnaires. The quality of the result is necessarily limited by the validity and completeness of the users' understanding of their own behavior, and by their ability to verbalize that information. Although the users may be quite expert at performing their job, they are rarely expert at analyzing or describing how they do the job, its information requirements, etc. Even an unusually insightful and verbally fluent user may produce a misleading characterization of a task's information requirements because of subjective biases. Indeed, such biases may easily result from the very experiential process by which the user acquires expertise. For example, often-repeated procedures may become "automatized" so that they cease to require constant attention. After a time, the user may be completely unaware of these processes and the information required to initiate them. In some cases, the most important elements of the user's behavior may be the very ones which the user is least able to verbalize without assistance. Furthermore, surveys of user requirements may be biased by the users' preconceptions concerning the capabilities and current uses of computer systems.

These problems are alleviated only slightly through the use of interviews (discussed in Figure 6). Even highly structured interviews are heavily dependent on the ability of the user to describe the task, information requirements, etc. While interviews with users are a good

<table>
<tr><th>Approach</th><th>Comments</th><th></th><th>References</th></tr>
<tr><td>Use of questionnaires to obtain ratings of the relative importance of various categories of information and system features.</td><td>Inexpensive. Difficult to be specific enough for detailed design decisions. Requires prior knowledge of all relevant information categories, although Delphi techniques might avoid this requirement.</td><td rowspan="5">An overriding limitation of questionnaire and survey techniques is their dependence on the correctness of the user's perception and description of the job, information requirements, work habits, etc. The user is expert at doing a job, not describing it. These subjective ratings often fail to correspond to objective performance comparisons, and may contain significant, if unintentional, biases and oversights.</td><td>Martin et al (1973)*
McKendry et al (1973)</td></tr>
<tr><td>Use of questionnaires to obtain estimates of time spent on each task associated with recipient's job.</td><td>Self-estimates of time spent on work activities are notoriously poor. If only relative time is required, this may be adequate.</td><td>Dever (1972)*</td></tr>
<tr><td>"Repertory Grid Technique", a questionnaire-based technique for determining user's "cognitive frame of reference".</td><td>Difficult to use successfully. High-level, and may not easily be made specific enough for detailed design decisions. Might be more useful for "personalized" systems than for capturing requirements of broad user class.</td><td>Peace & Easterby (1973)*</td></tr>
<tr><td>"Delphi Technique", a survey technique in which recipient's responses are fed back, anonymously. Recipient responds again, while aware of previous responses of entire group.</td><td>Relatively expensive. Promotes consensus and identification of all information categories, but may suppress important individual differences. No instances found of application in specific area of computer systems design.</td><td>Linstone & Turoff (1975)</td></tr>
<tr><td>"Policy Capture", one of several techniques for developing quantitative relationships between perceived system desirability and specific system features. In this case, relationship takes the form of a multiple-regression equation.</td><td>Somewhat expensive. Mathematical assumptions may be inappropriate. Paired-comparison procedure limits dimensionality. No instances found of application in specific area of computer systems design.</td><td>Stewart & Carter (1973)
(for discussion of other such techniques, see Cochrane & Zeleny, 1973)</td></tr>
</table>

Figure 5. Requirements Analysis: Quesionnaire and Survey Methods

Approach	Comments	References
Interviews with users to determine information requirements, decision points, organizational constraints, etc.	Used more or less universally. Formal discussion in literature is mostly in the context of management information systems. Has many variants, such as "structured" interviews. Although a skilled interviewer can overcome some of the limitations of subjectivity and inability of users to verbalize their practices, these limitations are still significant. To apply this method at the detailed system design level requires an insightful user, or interviewer, or both. Most useful for preliminary data collection.	Bentley (1976) King & Cleland (1975) Parker (1970)
"Ad Hoc Working Group", in which subject-matter experts devise system requirements by analysis and negotiation.	Appears somewhat effective at very high (undetailed) level. Has problems of subjectivity, and is susceptible to bias due to interpersonal relationships of group members (e.g., undue influence of high-status member). Probably irrelevant at detailed system design level.	Vandersluis (1970)*
"Critical Incident Technique", in which users are asked, via interview or survey, for information about incidents of particular success or failure in the process of which the computer system will be a part.	A broadly useful technique which often yields significant insights into critical functions and information. Often used by human factors personnel, but not evident in the computer systems design literature.	Flanagan (1954)
Job analysis techniques, such as task analysis, link analysis, and activity analysis, which attempt to characterize user behavior on the basis of direct observation.	Readily applicable to manual and clerical tasks, in which direct observation yields necessary raw data. Much more difficult to apply to cognitive tasks. Nonetheless, these techniques are broadly useful. "Task analysis" is often employed informally with inadequate detail and without necessary training in the technique. See also section on "Tasks and Task Properties".	Galitz & Laska (1969, 1970)* Wylie et al (1975)* (for general discussion, see also R. M. Miller, 1962)

Figure 6. Requirements Analysis: Interviews and Field Observation

source of preliminary information, they provide only part of the story. One beneficial function of a design guide might be the familiarization of the designer with more objective requirements analysis methods based on field observations, simulation and gaming.

Direct observation of user behavior during actual job performance is the simplest and most basic of these approaches. Standard job analysis techniques, such as task analysis, link analysis, and activity analysis, can provide structure to such field observation and suggest what to look for. These techniques are discussed in standard human factors texts, but are rarely described, in any detail, in documents dealing specifically with computer systems design. With appropriate instruction, system designers could employ at least some of these techniques effectively.

There are some limitations to these techniques, however. First, they involve observation of the user's existing work practices, which may not correspond to those involved in the use of the proposed computer aid. It is easy to fall into the trap of designing a computer system which enforces current -- and perhaps inappropriate -- manual practices. Second, the techniques are most suited for observation and analysis of manual or clerical tasks, in which the relevant user behavior is overtly observable. Cognitive, or problem-solving, tasks are difficult to interpret validly by simple observation.

On the surface, task analysis appears to be a simple, descriptive technique. To be useful, however, task analyses must describe behavior in terms of units of activity which are psychologically meaningful, with regard to both level of detail and type. This implies that sensitivity to, and probably training in, behavioral science is a prerequisite to effective use of task analysis. This prerequisite is often overlooked, and this failing may account for many unsatisfactory applications of the technique. It is not clear that a design guide can convey enough information to allow a "psychologically naive" system designer to employ this approach well. If task analysis is described along

with appropriate caveats, however, it might well be employed more effectively than at present, and the interested designer might be made aware of a need for additional training. It should be noted that a design guide emphasizing a single class of system might deal with this problem by presenting a description of candidate tasks known to occur in such systems, provided appropriate task-analytic research has been done.

Unlike field observation techniques, simulation and gaming approaches (Figure 7) allow the designer to investigate user behavior in novel situations. As we learn more about automated problem-solving and decision-making aids, it is becoming apparent that optimal aided approaches may differ drastically from the practices of the unaided user. Simulation methods not only allow empirical investigation of user interaction with not-yet-existent automated aids, but also provide a mechanism for the study of user behavior and information requirements in critical situations which occur only rarely in the actual application environment.

The simplest and most basic simulation method is "paper" simulation, in which the user interacts with a nonexistent system -- hand-simulated by humans using whatever computational aids are available and appropriate -- while solving one or more representative problems. A particular advantage of this approach is its compatibility with very unstructured simulations. By allowing the user to engage in aided problem solving with minimal constraints on both the dialogue and the types of information which can be requested, it is often possible to gain significant insight into the user's natural approach to the task (see Cross, 1969; Ramsey, 1975). A secondary benefit reported by Ramsey results from involving, in the simulation, personnel who will eventually be involved in the implementation of the system. These personnel acquire greater understanding of the user's problems, as well as possible implementation approaches, and are often more receptive to human factors inputs after such a simulation.

Approach	Comments	References
"Paper" simulation, in which the possible function of a computer system is simulated by human observers, in order to obtain information about the user's problem-solving and information-seeking behavior.	Relatively inexpensive, and often very informative. Can be used in very unstructured form, freeing user from dialogue constraints which might interfere with problem solving, or in very structured form to simulate exact properties of proposed system. Can be obtrusive, and requirement for manual computation can cause unacceptably slow simulation.	Cross (1969)* Ramsey (1975) Van Cott & Kinkade (1968)
"Protocol analysis", in which the user comments extensively on his activities during simulated problem solving, and formal content analysis of the resulting commentary ("protocol") is used to make inferences about user behavior and problem-solving processes.	Similar in many respects to paper simulation, but more obtrusive, more detailed. Transcription and scoring of protocols is very time-consuming, often restricts use to a small number of cases. This is the indicated approach when a detailed problem-solving model is the goal, but it has also been used, with reported success, in detailed dialogue design.	Carlisle (1973) Heafner (1975)** Mann (1975)
Interactive simulation or gaming, in which the actual system, or an interactive computer simulation of the system, is used with a contrived scenario to observe user behavior and system performance.	If done during initial design phase, this approach requires sophisticated software tools or expensive software development. It is very effective in identifying design defects and improving dialogue. Requires prior system design, and is, therefore, more appropriate for design evaluation and iteration than for initial inquiry into user requirements.	Lenorovitz & Ramsey (1977) Martin & Parker (1971) Proctor (1963) (for a broad discussion of gaming, see Shubik, 1975)

Figure 7. Requirements Analysis: Simulation and Gaming

"Protocol analysis" is a related technique which has become quite popular for the development of detailed cognitive process models, and is beginning to see application in computer system requirements analysis. In this technique, the user solves a simulated, or actual, problem while continuously reporting (either orally or in written form) the steps in the solution, their rationale, etc. This procedure can be highly obtrusive, in two ways. First, the requirement to report constantly on one's activities can interfere with the ability to perform the task, by causing the total workload to exceed the memory or processing resources of the problem solver. The situation is very similar to that in which a secondary task is imposed in order to determine the processing load associated with a primary task. Second, the requirement to report the nature and purpose of each step tends to emphasize the use of a systematic approach to the problem. It may very well cause the problem solver to commit to a particular strategy, or a particular "move", earlier than in an unconstrained setting. There is also a danger, with protocol analysis, that the emphasis in interpreting the results will be placed on the problem solver's verbal report rather than on the actual steps taken in solving the problem. There is experimental evidence that these do not always correspond (Best, 1977). In spite of these difficulties, however, and others mentioned in Figure 7, protocol analysis provides much more detailed information than other techniques, and has been used with reported success in connection with the design of interactive problem-solving aids and interactive languages.

Interactive simulation, in which the user interacts directly with a computer system to solve simulated problems, can be a very effective technique for gathering information about the user's behavior and information needs. This approach allows detailed simulation of the properties of the actual system (specific dialogue, response time, etc.). There are some obvious difficulties, however. Such a simulation can be done only after a fairly detailed -- even

if tentative -- design has been developed. Thus, interactive simulation is more appropriate for design evaluation and iteration than for initial user requirements data collection. In fact, the technique is usually applied only after the design is actually implemented, at least in skeleton form. The alternative -- simulation after the design is complete, but before implementation -- requires either significant resources or the use of specialized dialogue simulation tools which are not yet widespread (see Lenorovitz & Ramsey, 1977). There are many reports of beneficial uses of interactive simulation, however, (Parsons, 1972, briefly discusses several), and it should be considered an important tool in the overall design process. A later section discusses modeling of interactive systems, a topic closely related to interactive simulation. Many of the papers discussed there also bear on this topic.

In summary, there are several techniques for user requirements data collection and analysis which may be more objective and more effective than those in common use in the computer science community. Improved designer awareness of these methods, their advantages and pitfalls, and the procedures involved in their use, might be achieved by a design guide of the design-process-oriented type. It is not clear that a treatment of these topics could be achieved as effectively with the "reference handbook" approach, which usually assumes that the designer already has the data on which design decisions are to be based.

Interactive System Modeling Techniques

Once the designer has collected user and task data, it is sometimes advantageous, as the next step in the design process, to attempt to model the interactive system that is being designed. In the ideal case, a simulation model of the interactive system can be constructed. This would allow the designer to examine the effects of alternative design decisions before commiting to a final design structure. At the

very least, attempting to construct a model helps ensure that the designer has adequately analyzed user and task properties, since these are prerequisites for modeling efforts.

Depending upon the tasks that the interactive system is to be used for and the type of information that the designer is attempting to gain, one of several basic types of models could be used. These approaches differ in the degree to which they might reasonably be employed by an ordinary system design team to aid in the design process. In particular, some of the techniques depend primarily upon the adequacy of the task analysis, while others require considerable modeling sophistication and/or knowledge of human behavior. An overview of the major approaches to the modeling of interactive systems is presented in Figure 8.

The current literature on models of interactive systems can be summarized as follows. First, all of the types of models shown in Figure 8,with the exception of "models of human information processing," are directed toward specific task domains. Second, modeling efforts have been successful only when the modeler has a good understanding of a task domain and of user behavior in that domain. Third, models have been successful only in rather simple domains. Fourth, no generally applicable models exist and it is very unlikely that such models will be developed in the near future.

In the paragraphs below, we will consider each of the major approaches to modeling interactive systems. Following this, we will briefly consider how models of the user alone, rather than of the entire user-computer system, can be used to develop models of interactive systems. Finally, we will consider issues associated with validating models of interactive systems.

Network Models

A prerequisite for developing a network model is that the tasks performed by both the user and the system be describable in

Approach	Description	Comments	Principal References
Network Models	These models treat user and system as equivalent elements in the overall process. The individual tasks performed by both the user and the system are described in terms of expected performance and in terms of logical predecessor-successor relationships. The relationships define a network of tasks which is used as a performance model of the user-computer system. Such models are usually used to predict either the probability of failure or "success", or the completion time, of an aggregate set of tasks.	Network models allow performance data about user and computer system to be integrated in a single model even though original data came from a variety of sources. However, performance data must be provided for each task, as must rules for combining performance data from individual tasks to obtain aggregated performance predictions. This is often difficult because of questionable or lacking empirical data, and because performance interactions among tasks (especially cognitive tasks or tasks performed in parallel) may be very complex. Performance distributions are often assumed without data. In spite of these difficulties, the process of constructing such models is often a valuable source of understanding.	Baker (1970) Pew et al (1977) Siegel et al (1973)
Control-theory Models	These models are based on control theory, statistical estimation, and decision theory. The user is regarded as an element in a feedback control loop. Such models are usually used to predict overall performance of the user-computer system in continuous control and monitoring tasks.	Control-theoretic models are more quantitative than other performance models. They may address user-computer communication broadly, but they ordinarily do not deal with details of the interface, such as display design. Therefore, their utility as an aid to the interactive system designer may be limited. Not much work has yet been done in applying these modeling techniques to situations in which the main user activities are monitoring and decision making, with infrequent control actions.	Pew et al (1977)

Figure 8. Major Approaches to the Modeling of Interactive Systems (Page 1 of 2).

Approach	Description	Comments	Principal References
Decision-Theory Models	These models concern the decision-making behavior of the user. They require the specification of: (1) a set of possible states of the world, with their estimated probabilities, and (2) a set of possible decisions, or courses of action, which might be taken, together with their expected values and costs in the various possible states of the world. Considering the values and costs, together with the evidence of particular world states, a decision-theoretic model can select courses of action.	Decision-theoretic models can be used to suggest "optimal" decisions or to describe the observed decision-making behavior of users. In both modes, these models are frequently used in decision aids. If it is reasonable to describe user behavior in terms of such a model, these models can also be useful to the system designer, as by suggesting information required by the user. However, such models have thus far been developed only for relatively simple, two-alternative decision situations. Most real-world tasks are more complex, and gross distortions may result from any attempt to express them in this simple form.	Pew et al (1977) (for a discussion of the use of these models in automated decision aids, see later section on "Problem-Solving Aids")
Models of Human Information Processing	In general, these models involve a characterization of: (1) the task environment, including the problem and means of solution available, (2) the problem space employed by the subject to represent the problem and its evolving solution, and (3) the procedure developed to achieve a solution. The method used to develop such models involves intensive analysis of the problem to be solved and of protocols obtained from problem solvers during solution.	Ideally, such efforts might lead to an integrative model of human information processing useable in a variety of design applications. However, existing models are either too task-specific for this use or are insufficiently detailed. Furthermore, relationships between task requirements and human performance capabilities and limitations are inadequately understood for human information processing tasks. There are many good models applicable to very specific tasks. If they happen to be relevant to particular user tasks in the system undergoing design, and if the designer is sophisticated enough to recognize their relevance, these models may be useable, but this is asking a lot of the designer.	Mann (1975) Pew et al (1977) Ziegler & Sheridan (1969)
Computer System Models	These models attempt to describe the behavior of the computer component of an interactive system, but do not attempt to model user performance in detail. Some of the models do characterize user behavior in terms of the statistical properties of user commands for a particular application. The models usually attempt to predict such performance factors as system response time, CPU and memory loads, and I/O requirements.	These models tend to be relatively crude, but can be useful in determining whether or not user requirements with respect to response time and other gross system performance measures can be satisfied by a proposed system. They are of little assistance in determining what the user requirements are.	Carbonell et al (1968) Foley (1971) Grignetti et al (1971) Shemer & Heying (1969)

Figure 8 (Concluded).

terms of predecessor-successor relationships. It is also advantageous if the tasks are fairly independent. In this case, performance data can be collected on each task individually, with little concern for the performance effects of interactions among tasks. If there are interactions among tasks, the effects on performance on individual tasks are generally very complex, not easily predicted, and affected by a large number of situation-specific and user-specific factors. In this case, collecting performance data can involve a great deal of controlled, empirical work (cf., Norman and Bobrow, 1975).

Existing network models of interactive systems tend to be limited to tasks that have a linear ordering (e.g., Baker, 1970). Although a linear model clearly satisfies the requirement of having predecessor-successor relations, it assumes that there is a single path that must be followed from start to completion of a problem. More complex types of networks (e.g., hierarchies or heterarchies) could also be used. Such models more easily allow for the fact that the user-computer interaction can take a variety of paths.

An example of a hierarchical network model is a "procedural net" (Sacerdoti, 1975). A procedural net represents the possible solutions to a problem in levels, with each successive level involving more detail than its predecessors. For example, one procedure might specify "classify incoming messages," and other procedures, at lower levels in the hierarchy, would specify in more detail how this was to be accomplished. Predecessor-successor relations are defined between levels and also within levels. At a given level, the procedures are linked together by predecessor and successor relations that define a partially ordered sequence of operations necessary to carry out some higher-level procedure. Examples of procedural net models that allow for heterarchical, rather than strictly hierarchical relations, are presented in Brown and Burton (1978) and in Van Lehn and Brown (1978).

Although more general than strictly linear models, procedural net models are more complex to construct and have currently been developed

for only relatively simple types of tasks (learning mechanical assembly, Sacerdoti; learning arithmetic, Brown and Burton).

Baker's model (Baker, 1970; Siegel et al, 1973) is a particularly good example of a network model used to investigate the properties of a particular user-computer system. However, like any sophisticated model, it was expensive to construct. Furthermore, by modeling at a detailed level (a strength of the model) it achieves very limited generalizability to other systems or tasks. Such difficulties apply to all modeling approaches, and should be carefully considered before attempting model development.

Control-Theory Models

As their name implies, these models are best suited for control-type tasks. From a user performance perspective, control generally does not involve problem solving, in the sense that the user must actively search for a solution or resolution of the current task. Rather, this type of task generally involves fairly automatic application of well-learned algorithms and procedures. In this case, user performance variance is extremely low and statistical estimation procedures provide adequate descriptions of performance.

In control-theory models, the user is treated as an element in a feedback-control loop. Because both user and system performance variances are low, very good prediction of overall user-computer system performance can be obtained. A potential pitfall of such models is that treating the user as an element in a control loop may lead the designer to not explicitly focus on the properties of the user-system interface.

Decision-Theory Models

Decision-theory models are applicable to tasks in which the user must select from among alternative actions and evaluate the probable effects of applying these actions. The development of such models assumes that the criteria with respect to which actions are to be eval-

uated are well-specified and that a "correct" evaluation of each alternative action exists. Although these are serious restrictions, and limit the types of tasks to which decision-theory models are applicable, the successful development of such a model provides the designer with an important advantage. Decision-theory models can, with some modification, be incorporated into an interactive system as decision aids. A decision-theory model indicates not only what aids would be useful, but also indicates how they should be implemented (cf., Freedy et al, 1976). The designer should note, however, that the task involved must be understood to a fairly fine level of detail and this, as a consequence, limits such models to relatively simple situations.

Models of Human Information Processing

Although potentially the most general model, this type of model is also the most difficult to construct. The view of a human as an information processor indicates, in part, the influence of computer science concepts on psychology. In such models, it is not uncommon for the user's processing abilities to be represented in a manner similar to automata theory or Turing machine descriptions of computer systems.

The generality of such models results from the fact that an adequate model of human information processing could be applied to any task domain. Although there is a very large body of research in this area, current knowledge is still insufficient to supply all of the detailed information necessary to construct such a model. Such models have been developed, however, for well-specified tasks, for which it is not necessary to develop a complete model of the user as an information processor. If the designer can find an existing model that is applicable to the current system design, it may be advantageous to use it. If no such model exists, however, developing such a model is probably beyond the capabilities and constraints of most interactive system designers.

Computer System Models

These models are intended primarily to describe computer system behavior. In general, they give inadequate attention to, or completely ignore, user performance or the user-computer interface. These models are most useful in determining whether or not such user requirements as system response time are satisfied. They are also useful, as models of existing systems, for "fine tuning" overall system performance for simple, well-defined tasks.

Models of the User Alone

In Figure 8, we presented the major approaches to the modeling of interactive systems. A distinction needs to be made, however, between models of interactive systems, as such, and models of either the user or *the computer system in isolation. Even in a given approach* to modeling interactive systems, differential attention can be given to either the user or the computer system.

Computer system models are near one end of this continuum; they focus largely on the computer system and give little attention to the user. The remaining approaches shown in Figure 8 were originally developed as models of human (and thus user) performance and have gradually been adapted to include computer system performance. That is, the assumptions underlying these approaches were derived from research on human behavior. The development of such models into models of interactive systems is due, in part, to the recognition that some types of unaided human behavior are also observed in human behavior with interactive systems.

There are a large number of models of the user alone that have not yet been adapted as models of interactive systems. Such models are generally represented as simulation models of human performance in specific task domains. Since the intent is to develop viable psychological models of behavior, these models tend to be extremely detailed.

Many models of the user describe human performance in terms of the memory and process limitations of the human information processor. If the designer can find an appropriate model, then the interactive system may be designed to overcome, at least partially, the effects of these limitations. In general, however, the amount of detail required in such models limits their development to relatively simple tasks, which may be of little interest to interactive system designers. Further, it may take a great deal of sophistication, on the part of the designer, to translate the psychological terms and concepts of such models into a form suitable to express a system design.

Model Validation

In general, validating a model requires that the model produce predictions of system performance that can be compared with observed system performance. Since the basic purpose of such models is to provide the designer with a tool for exploring design alternatives before committing to a design structure, however, such validation is ordinarily impossible. In this case, model validity is a function of the adequacy of the user and system data that were collected and the appropriateness of the chosen model for a specific type of task. Without sufficient data on user and system requirements, model construction can be a futile exercise. However, attempting to construct a model may point out inadequacies in the data collection and analysis process and, as a result, lead the designer to consider requirements that were neglected before.

Conclusions

As was indicated above, there are a large number of constraints on the development of models of interactive systems. The selection of a modeling approach must be based on the task domain involved. Further, the designer must have a good understanding of this task domain and currently successful models are restricted to rather simple domains. As a result, no generally applicable approach to developing models of interactive systems exists.

It has been suggested elsewhere in this report that restricting the application domain of a design guide might allow a more explicit and detailed treatment of the tasks involved. This may also allow a more useful discussion of existing models and modeling techniques applicable in the chosen application area. By concentrating on a particular kind of interactive system, the design guide might even be able to suggest a basic model as a starting point.

Task Allocation

In the "traditional" approach to the design of person-machine systems, the next step after requirements data collection is an allocation of the identified tasks to either the user or the system. This classical "task allocation" method is discussed very well by Chapanis (1965). It is implicitly assumed, in this method, that the individual tasks identified by (for example) task analysis are highly independent. Thus, any one of these tasks can be assigned to either the user or the system with little impact on the performance of other tasks. Typically, each task is analyzed to determine which system component is best suited to the task, and the task is assigned accordingly.

For some kinds of systems (e.g., manual control, simple clerical systems), this method works reasonably well, but it has deficiencies which make it inappropriate for most modern interactive computer system design efforts. The most effective designs are often those which use the computer to assist and complement the user in difficult tasks, rather than simply assigning the mechanical tasks to the computer (see Jordan, 1963, Vaughan & Mavor, 1972). In fact, interactive aids often allow the task to be approached in a way which is impossible for the unaided user. There is nothing in the traditional task-analysis/task-allocation method which helps the designer recognize the need for -- or nature of -- a novel interactive approach to problem solving. If the method is employed uncritically, it may simply lead to automation of an existing, inferior approach.

This is a significant pitfall for the conscientious designer who is not well educated in human factors. Because this method works well in many systems which do not involve interactive computer aids, it is heavily emphasized in many standard human factors texts. The uncritical use of such methods can place the designer in some peril.

A design guide should certainly discuss the use of classical task allocation, but should also outline the limitations of the method and help the designer identify those circumstances under which it should be applied.

Unfortunately, the alternatives available to the designer are more difficult to identify. If, as we have suggested, design is primarily an analytic-synthetic process, the designer's experience is necessarily a dominant factor in recognizing the applicability of particular types of interactive aids. A design guide might assist this process by providing examples of different types of aids, along with suggestions about the types of tasks in which they are appropriate. This is discussed in the next section.

PROBLEM-SOLVING AIDS

In this section, we present a survey of the types of aids that have been used for man-computer problem solving and some heuristics, or "rules of thumb" for selecting an appropriate aid for a given situation. Since the usefulness of any aid is ultimately determined by the task to which it is applied and the problem-solving behavior that is elicited by that task, we will begin by describing common types of problem-solving behavior. Following this, we will consider what specific aids are likely to be most useful for each type of behavior.

"Man-computer" problem solving is defined as a type of problem-solving activity that involves both a human and a computer component. "Human problem solving", of itself, can be characterized as an "informa-

tion processing" task. The human has a set of cognitive abilities and mechanisms that can be applied to a variety of tasks. The extreme versatility that can be observed in human problem solving is due to the fact that these mechanisms and abilities can be modified, within limits, and ordered in different ways. Although versatile and adaptive, the human problem solver also is subject to fairly severe memory and processing limitations. That is, human memory may be insufficient to deal with all of the information that is generated by some process. Or a process, although its method of execution is well known, may require more processing resources than the human has available, perhaps because of competing processes. "Man-computer problem solving" refers to a situation in which the computer is used to effectively extend the human's information processing resources and reduce the effects of these limitations.

In this review, we will consider a problem to consist of a set of "knowns" and a set of "unknowns" (cf, Greeno, 1973). In a geometry problem, for example, the "knowns" are the initial situation and the theorems and axioms that can be applied to it,and the "unknowns" are the theorems and proofs to be derived. "Problem solving" refers to the process of generating a connection between the "knowns" and "unknowns". Although this definition is extremely broad and appears to encompass a large number, if not all, human activities, it is important to note that not all tasks are considered to be problem-solving tasks.

In determining whether a given task is or is not a problem-solving task, it is necessary to determine whether the connection between the "knowns" and "unknowns" is <u>generated</u> or <u>recognized</u>. In story or scene understanding, for example, there are "knowns" and "unknowns". In story understanding, we may be presented with a set of characters ("knowns") who are performing certain actions in order to achieve some goals ("unknowns"). The connection, however, is presented in the story and the reader need only <u>recognize</u> it; it is not <u>generated</u>.

Although story comprehension is perhaps the clearest example of a task that involves recognition rather than generation, other tasks also have a predominant recognition component. Examples include signal detection, surveillance and monitoring tasks, or any task in which the principal activity involves noticing that some clearly coded event occurs or is present.

All of the aids considered in this section are concerned with tasks that primarily involve generation. Aids for recognition tasks, which typically involve display coding or time-compressed displays, are discussed elsewhere in this report.

Ideally, a system designer should be able to select appropriate problem-solving aids on the basis of the problem-solving task involved. For example, if the system is designed for control tasks or logistics tasks, then specific aids could be apparent. Unfortunately, for two reasons, this is not the case. First, although we can define problem solving as a generation task, it must be noted that problem solving is not a unitary task, but typically involves a variety of subtasks. In a given problem-solving situation, one or more of these subtasks is generally more crucial than the others. A description of each of these subtasks is presented in Figure 9. The problem-solving behavior that is most appropriate for a given task is determined largely by the subtasks that are most important to successful performance rather than by the overall task itself.

Both the specific task and the experience of the user, therefore, affect the type of problem-solving behavior that will be employed. In selecting problem-solving aids, it is necessary for the designer to consider the type or types of behavior that will be exhibited. In a separate section of this report, we described methods for requirements analysis. If the designer is primarily concerned with a man-computer problem-solving system, then this analysis should include consideration of the type(s) of problem-solving behavior involved.

Subtask or Phase	Description	Comments	Principal References
Problem Recognition	The first stage in problem solving is to recognize that a problem exists. People are frequently slow to recognize, or at least react to, problems. This is especially true in situations in which a person must monitor the current state of the environment and detect or react to critical changes.	A primary need is for an aid that alerts the problem solver to "relevant" changes in the environment. The relevant variables for a given task can be difficult to define. Current status displays, historical displays, and aids for dealing with degraded data can be useful. If the relevant variables are identified, coding techniques can be very useful.	Booth et al (1968) Chesler & Turn (1967) Scanlan (1975)** Smith, R. L. et al (1972) Topmiller (1968) Wylie et al (1975)
Problem Definition	After a problem is recognized, the problem solver must determine how to formulate, or represent, the problem. In most cases, there are several alternative formulations for a given problem. The overall success of problem solving strongly depends on selecting an appropriate formulation.	Aids that provide a change in problem representation (e.g., graphical displays, isomorphic representation) can be extremely useful. Developing alternative representations requires a thorough understanding of the specific problem and the problem-solving processes that are most appropriate. Allowing the problem solver to decompose a problem into subtasks and recombine these subtasks in various ways can be useful in problems with relatively independent tasks. This type of aid is less difficult to develop than changes in problem representation, but it is also less general.	Balzer & Shirey (1968) Cushman (1972) Krolak et al (1971) Newsted & Wynne (1976)* Smith, H.T. (1974)* Stewart (1974)
Goal Definition	In some cases, the goal to be achieved is pre-defined. In other cases, (e.g., tactical planning) the problem solver must select an appropriate goal. A selected goal must be not only appropriate, but also attainable.	The primary difficulty is that a selected goal may not be attainable. It may be useful to aid the problem solver in generating several alternative, logically consistent goal structures and to delay selecting a specific goal until later in the problem-solving process. Research on goal definition is lacking.	None
Strategy Selection	Strategy selection is concerned with determining the general approach that will be used in problem solving. In some cases, a certain strategy is dictated by the problem representation that is selected. In general, strategy selection is based on previous experience with a given class of related problems.	The majority of strategy-selection aids are concerned with specific problem domains. This is appropriate since strategy selection is strongly driven by experience in a given domain. In domains in which problems can be decomposed into fairly independent subproblems, aids that allow the user to select strategies for these subproblems independently before combining them into an overall strategy can be very useful. Additional research is needed on the nature of specific problem-solving tasks and the strategy selection heuristics used by expert problem solvers. This would enable the construction of techniques to aid the less experienced user in this phase of problem solving.	Bennett (1971) Caruso (1970)* Wilde (1969)*

Figure 9. Basic Subtasks or Phases Involved in Problem Solving (Page 1 of 2).

Subtask or Phase	Description	Comments	Principal References
Alternative Generation	In well-defined tasks, the problem solver can usually generate all alternative actions that may be appropriate. If there is a large number of alternatives, however, the problem solver may not be able to retain all alternatives in memory for later evaluation. If the task is not well-defined, the problem solver may not be able to generate appropriate alternative actions.	Aids that store a large number of user-generated alternatives can easily be developed and can also be effective. The principal need is for aids to suggest alternatives that the user is unable to generate. Such aids have been developed for training applications and for cases in which the computer has been programmed to generate optimal solutions without explicit user interaction. For ill-defined task environments, aids that suggest hypotheses to be tested may aid in alternative generation. Although potentially very useful, such aids could be difficult to construct.	Brown et al (1974) Carlson & Hodgson (1977)* Gagliardi et al (1965)* Hormann (1967)
Alternative Evaluation	Problem solvers are generally very good at evaluating alternatives in a manner consistent with their perception of the problem and the goal to be achieved. If the alternatives have far-reaching consequences or if they must be evaluated with respect to a large number of factors, the problem solver's memory and processing limitations may be exceeded.	In extremely well-defined task environments, aids have been developed that allow the user to simulate the consequences of various alternatives. Although they are very useful in specific cases, such aids have limited generality. Aids that capture the user's evaluation heuristics and then filter information to be consistent with these heuristics and sometimes even present alternatives considered to be optimal are especially useful when a large number of evaluation heuristics must be applied. This type of aid is both effective and general, but it requires a great deal of effort to implement.	Balzer & Shirey (1968) Brown et al (1975) Davis et al (1975)** Doutriaux (1973)* Freedy et al (1976)** Michie et al (1968)** Rapp (1972)* Smith, H.T. & Crabtree (1975)** Yntema & Clem (1965)*
Alternative Selection and Execution	The last phase of problem solving is concerned with implementing the solution.	Automatic execution of user-specified actions can aid the user in interacting with the problem-solving environment. Aids that automatically take over the problem-solving process may also be useful, but they should be used with caution.	Bursky et al (1968) Freedy et al (1976)** Hanes & Gebhard (1966)* Pulfer (1971)

Figure 9 (Concluded).

For the purposes of this review, we will consider three dimensions along which behavior can be classified. Although this is a simplified view of problem-solving behavior, it seems to be an appropriate level of detail. Finer classifications of behavior, though possible, would tend to obscure the commonalities between behavior in various task domains and the aids most appropriate for these behaviors.

Abstractions -- No abstractions

Problem solving can be viewed as an activity that takes place within a "problem space" (cf., Newell and Simon, 1972). A problem space represents the problem solver's view of the problem to be solved, the actions that can be taken to search for a solution, etc. The specific elements in the problem space may or may not closely correspond to elements in the environment in which the solution will be implemented.

In problem solving by abstraction, the problem is reformulated into higher-level, more general terms. This is most useful in large, complex tasks in which the problem elements that (eventually) need be considered would exceed the problem solver's memory resources. Software design is an example of a task that involves abstraction; a designer initially works with an abstract representation of the design, rather than with the algorithms or code that will eventually implement the design. After an abstraction is formed, problem-reduction heuristics are generally applied to partition the original problem into a series of (hopefully) independent subproblems, which can be dealt with at the level of detail required by the final solution.

In problem solving with no abstractions, all actions that the problem solver may generate are directly related to the problem-solving environment. That is, all actions could be executed immediately. This criterion is useful in deciding whether a problem involves abstractions or not. In problem solving by abstraction, the effects of considered actions must be simulated by the problem solver, since they cannot be directly applied.

In problem solving by abstraction, the primary subtasks involved are problem definition and strategy selection. Since abstraction implies that the problem will be reformulated, it is important that an appropriate problem definition be used. Strategy selection, in this case, refers to the choice of an appropriate problem reduction. In general, problem solving by abstraction is effective only if the problem can be reduced into a series of subproblems.

When no abstractions are used, the focus is on alternative generation, alternative evaluation, and alternative execution. Since the actions that can be taken will directly affect the environment, appropriate actions must be considered (alternative generation) and the effects of actions must be known before the action is applied (alternative evaluation). Because actions can directly affect the environment, successful problem solving requires a successful interaction with the environment, which is determined by the alternative execution subtask.

Search -- No search

Search behavior implies that a solution to the current problem is sought by searching through a set of possible solutions. This type of behavior generally involves the application of inference rules or heuristics to newly presented or generated information in order to test whether a solution is achieved. Although the problem solver may know all possible solutions, it is more likely that only the inference rules that test for a solution (and conceivably could generate all possible solutions) are known. No-search behavior implies that previously developed solutions (or parts of solutions) are used in problem solving. The problem solver recognizes that the problem (or part of the problem) has been solved before and an appropriate solution is retrieved. It may frequently be the case that the retrieved solution needs some modification before it can be applied, but it is, in most important respects, at least an appropriate "outline" for the required solution.

Search behavior primarily involves three subtasks. Problem definition leads to the selection of inference rules and determines the possible solutions that will be considered. The strategy selection subtask determines how the selected rules will be applied. Similarly, the generation of feasible solutions can be considered as alternative generation.

In no-search behavior, the primary focus is on alternative evaluation and alternative execution. Alternative evaluation refers to the need to evaluate the appropriateness of the alternative, retrieved solutions, and alternative execution is concerned with mapping the retrieved solution onto the problem-solving environment.

Design tasks are examples of situations in which no-search behavior is frequently observed. In engineering design, for example, an experienced designer is well aware of the possible structures that represent valid designs, and is generally able to retrieve a previously developed design and modify it to fit the current problem. In tasks in which there is not a great deal of commonality or similarity among problem solutions, search behavior is typically observed. Scheduling or resource allocation in highly dynamic environments are examples of such tasks.

Data-driven -- Conceptually-driven

Data-driven behavior is guided by particular aspects of the task domain. That is, certain domain-dependent aspects of the problem suggest "crucial" steps in problem solving or suggest necessary problem-solving processes. Command and control and medical diagnosis are examples of such tasks. Conceptually-driven behavior, on the other hand, is guided by the problem solver's previous experiences with similar domains. Most design tasks are in this category.

In data-driven behavior, problem recognition and alternative evaluation are the most crucial problem-solving subtasks. Problem recognition is important since salient, relevant aspects of the environ-

ment must be recognized and reacted to. Alternative evaluation is important since this type of behavior focuses on detailed aspects of the environment and the effects of actions at a low level of detail must be well understood.

In conceptually-driven behavior, the primary subtasks are problem definition, goal definition, and strategy selection. Since conceptually-driven behavior is guided by previous experiences, the problem definition subtask should result in the problem being formulated so that previous relevant experiences will be retrieved. Since these retrieved experiences are frequently goal-oriented -- that is, selected on the basis of the goal that is desired -- goal definition is crucial. In addition, the goal definition subtask should ensure that the selected goal is both appropriate and attainable. Strategy selection determines how the retrieved information will be applied.

Types of problem solving aids that have been proposed and developed, to varying degrees of detail, are described in Figure 10. The reader should note that only general descriptions of aids are provided and the exact form that an aid will take upon implementation may depend on the specific task domain and the type of behavior involved. For example, strategy selection for problem solving involving abstraction may suggest problem-reduction heuristics, but if search behavior is involved, an interactive aid might suggest which inference rules to apply.

Figure 11 indicates which aids are most likely to improve performance as a function of the type of problem-solving behavior involved. Only those aids that are considered to be most useful for a given type of behavior are indicated. Aids other than those indicated could also be used, but their effects would be expected to be, in most cases, less significant. Some of the aids listed in Figure 10 are not associated with a particular behavior type. The reader is referred to Figure 10 for comments on the use of these aids.

Aiding Mechanism	Description	Comments	Principal References
Alternative Evaluation	These aids may either automate the user's evaluation criteria, require the user to use established criteria, or simulate the results of actions that do not have well established evaluation criteria.	Except for aids that automate the user's evaluation criteria, these aids are task-specific. Most useful if the task is not well-defined or if a large number of evaluation criteria need be considered.	Brown et al (1975) Hormann (1967) Rapp (1972)* Smith, H. T., & Crabtree (1975)**
Alternative Generation	These aids are primarily used to generate alternatives that the user would not normally consider or, for extremely well-defined tasks, to present algorithmically determined alternatives.	Except for well-defined task domains, where they may have very little impact, they are difficult to construct. Can be cost-effective for training applications, but generally are of limited use in complex problem-solving tasks.	Baldwin & Siklossy (1977) Gagliardi et al (1965)**
Automatic Action Execution	Such aids permit the user to name the desired action without explicitly carrying out the steps involved in its execution.	Most useful when the results of applying an action do not impact subsequent problem-solving actions. If this is the case, the user may need sophisticated alternative evaluation heuristics.	Carlson & Hodgson (1977) Hanes & Gebhard (1966)* Pulfer (1971)
Automatic Takeover	This type of aid functions as an automated decision maker that is able to select alternative actions on the basis of prior observations of the human decision maker's behavior. Although allocation of control to this aid occurs automatically, whenever some criterion of correspondence between predicted and observed human behavior is reached, voluntary turnover of control is also possible.	Although demonstrated to be effective in some contexts (e.g., control tasks), the range of tasks in which this is appropriate is not well understood. User acceptance may be low and should be carefully examined.	Freedy et al (1972) Steeb & Freedy (1976)*
Backtracking	Such an aid allows the problem solver to "undo" the effects of recent actions and return to an earlier state of the problem-solving process without actually starting over.	Useful in tasks where it is possible to "undo" recent actions. Can improve performance at relatively little development cost.	Carlson & Hodgson (1977) Michie et al (1968)* Teitelman (1972)*
Better Weighting of Unreliable Data	This aid re-codes low-fidelity data into a form that is more readily useable by the problem solver.	Depends on the ability to accurately recode low-fidelity data.	Topmiller (1968)** Howell & Gettys (1968)*
Change of Problem Representation	Typical implementations of this aid present problems as isomorphic variations of more standard problem representations. It is intended that this will aid the problem solver in selecting an appropriate and efficient problem formulation.	Most useful in well-understood tasks. An inappropriate representation may seriously degrade performance.	Chesler & Turn (1967) Smith, H. T. (1974)** Newsted & Wynne (1976)*

Figure 10. Types of Problem-solving Aids (Page 1 of 2).

Aiding Mechanism	Description	Comments	Principal References
Decision Consistency Improvement	This type of aid assists the users applying their own decision strategies consistently in cases in which these strategies are complex.	Useful for expert problem solvers in well-defined tasks. Including sufficient versatility to adapt to individual users may be difficult.	Davis et al (1975)** Freedy et al (1976)**
Decision Strategy Improvement	Such aids assist the user in applying problem-solving techniques that would not normally be considered or known.	Useful in well-defined tasks in which optimal, or near optimal, problem-solving techniques are known, or in tasks in which general heuristics, such as problem reduction, are applicable. Requires detailed knowledge of the task.	Caruso (1970)* Gagliardi et al (1965)** Rogers et al (1964) Wilde (1969)*
Decomposition and Recombination	This type of aid allows the user to divide the original problem into subproblems. The solutions of the various subproblems are then combined into a solution to the original, larger problem.	Useful only if a task can be decomposed into independent subproblems. Requires a good understanding of the task.	Krolak (1971)*
Disruption of Psychological Set	Such an aid is intended to disrupt any bias or "sets" that the user may employ and, thereby, stimulate more creative, or novel, problem-solving attempts.	Potentially useful, but may disrupt an <u>appropriate</u> "set".	Stewart (1976)
Extended Memory	This aid allows the user to store and retrieve problem-relevant information. This information may initially be generated by the user or by other problem-solving aids, such as aids for alternative generation and evaluation.	Very useful in almost all tasks. Success is related to the ease of retrieval from external memory.	Balzer & Shirey (1968) Newsted & Wynne (1976)* Smith, H. T., & Crabtree (1975)**
Lockout	In an interactive problem-solving situation, this technique restricts the problems solver's access to the computer for some period of time after the presentation of the results from the current request for information.	Although demonstrated effective in some contexts, user acceptance was low. The tradeoff between user performance and user acceptance should be carefully considered.	Boehm et al (1971)** Seven et al (1971)**
Rapid Trial-and-Error	This aid allows the user to rapidly and easily examine the consequences of alternative action by simulating their application.	Easily implemented in well-defined tasks. May offset inadequacies in decision strategy improvement aids.	Balzer & Shirey (1968) Carlson & Hodgson (1977) Rapp (1972)* Wilde (1969)*
Strategy Capture	These aids attempt to model and predict the user's behavior. Strategy capture is generally used in conjunction with other aids, such as automatic takeover or alternative evaluation.	A prerequisite for developing automatic takeover aids. Best suited to tasks that allow algorithmic, rather than heuristic, strategies.	Doutriaux (1973)* Freedy et al (1976)**

Figure 10 (Concluded).

Aiding Mechanism \ Behavior Type	Abstraction	No abstraction	Search	No search	Data-driven	Conceptually-driven
Alternative evaluation		X	X		X	
Alternative generation		X	X			
Automatic action execution				X		
Automatic takeover						
Backtracking		X	X			
Better weighting of unreliable data					X	
Change of problem representation	X					X
Decision consistency improvement					X	
Decision strategy improvement	X					X
Decomposition and recombination	X					
Disruption of psychological set						
Extended memory		X	X	X		X
Lockout						
Rapid trial-and-error		X	X			
Strategy capture					X	

Figure 11. Relationship of Aiding Mechanisms to Problem-solving Behavior Type.

The greatest difficulty involved in making our knowledge of problem-solving aids useful to the designer is the relatively abstract level of that knowledge. If design guidelines were to be written for a very limited class of systems, it might be possible to suggest very specific aids. For more general guidelines, it would be necessary to greatly simplify the language in which problem solving has just been discussed, and provide guidance to help the designer recognize the kinds of problem-solving behavior involved in the particular tasks at hand. Furthermore, it would probably be necessary to provide very explicit examples of the various types of aids which are known. This all appears feasible, but is by no means easy, and we are unaware of any such guidelines which have been developed in the past. If designers operate primarily by synthesizing designs from already known elements, such a presentation might greatly enhance the designer's utilization of aiding techniques which happen to lie outside the designer's past experience.

INTERACTIVE DIALOGUE

After the designer has settled upon a basic set of functions to be performed by an interactive system, the next major step is the determination of the nature of the interactive dialogue. There are a number of fundamentally different types of dialogue, and a very large number of properties which an interactive dialogue may exhibit. A systematic treatment of this topic seems to require that the basic properties and types of dialogue be discussed independently of such factors as the details of the command language, display formatting, selection of display and input devices, etc. A separate discussion of basic issues is particularly appropriate since there is much less explicit guidance available, in the human factors literature, with respect to selection of a dialogue type than with respect to the detailed design of the dialogue after the type is determined.

This section discusses interactive dialogue in general, and provides a more detailed treatment of several common dialogue types. Questions of display formatting, display coding, and display and input device selection are reserved for later sections.

BASIC DIALOGUE PROPERTIES

Prior to a discussion of various dialogue types, it is appropriate to consider a few basic properties on which interactive dialogues may differ. Figure 12 contains a summary discussion of some of the most important properties. Many other aspects of interactive dialogue could be included here, but the listed properties are particularly important, apply to all dialogue types, and have at least some empirical data which might be of use to the designer.

Initiative

"Initiative" is probably the most basic interactive dialogue property. In computer-initiated dialogue, the user inputs commands,

Dialogue Property	Description	Comments	Principal References
Initiative	Initiative is concerned with whether the user or the computer initiates the individual information transactions within the dialogue. If the computer asks questions, presents alternatives, etc., and the user responds, the dialogue is "computer-initiated". If the user inputs commands without such computer "prompting", the dialogue is "user-initiated". "Mixed initiative" and "variable-initiative" dialogues are also possible.	Computer-initiated dialogues are preferrable for naive users or trainees, and for casual users. Computer-initiated dialogue allows reliance on passive, rather than active, vocabulary, implicitly teaches the user a "system model", and allows use of the system by a user who has not yet internalized such a model. Computer-initiated dialogue is also satisfactory for experienced users if use involves few transactions or system response is very fast. The latter usually implies a "smart" terminal. A slow, computer-initiated dialogue is very disruptive to the frequent, experienced user. See the later section on "response time". For most systems, designers should consider allowing the user to select either dialogue mode.	Johnson (1977) Martin (1973) Thompson (1969, 1971)
Flexibility	Flexibility is a measure of the number of ways in which a user can accomplish a given function. High flexibility can be achieved by providing a large number of commands, by allowing the user to define or redefine commands, etc.	There is evidence that nonprogrammer users adopt a "satisficing" strategy with respect to flexible dialogues. That is, they tend to utilize known methods for solving a problem even when the system provides less cumbersome methods, known but not yet learned by the users. There is also evidence that more flexible dialogues degrade performance (especially by increasing error rates) of relatively inexperienced users. Thus, highly flexible dialogues may be undesirable except for experienced and/or sophisticated users.	Eason (1976) Stewart (1976b)* Walther (1973)** Walther & O'Neil (1974)**
Complexity	Complexity is related to flexibility. Complexity is a measure of the number of options available to the user at a given point in the dialogue. Low complexity can be achieved by using few commands, or by partitioning the commands so that the user selects from a small set at any given time.	The effects of dialogue complexity on performance are unclear. It seems reasonable to expect that there is some optimal level of complexity for a particular task and user type, with degraded performance resulting from significantly more or less complex dialogue structure. There is evidence which suggests that a large number of redundant or irrelevant commands impairs user performance, and that extreme "simplification" of the dialogue by hierarchic structuring is also detrimental. Complexity appears to interact with user properties, but a systematic description of this interaction is not yet possible.	Baker & Goldstein (1966)** Boies (1977) Carlisle (1974)**

Figure 12. Some Major Properties of Interactive Dialogues (Page 1 of 2).

Dialogue Property	Description	Comments	Principal References
Power	Power is the amount of work accomplished by the system in response to a single user command. In a dialogue with powerful commands, the user may accomplish, with a single command, an operation which would require several commands in a system with less powerful commands. Power is related to flexibility and complexity.	Obviously, the power of the commands must somewhat correspond to the user's needs, which may vary over time and users. In some application areas, a large range of command power is possible and use of relatively high-power commands can be very effective (e.g., matrix operators in a mathematical system). If power is achieved by using powerful commands and facilities instead of less powerful, but more basic commands, the result is a reduction in generality of the system. The MICA study found this to be a significant factor in system rejection by both managers and technical personnel. On the other hand, provision of powerful commands in addition to a more basic set tends to increase dialogue complexity. One possible solution is to partition the dialogue so the less sophisticated user is exposed to a subset of commands.	Eason (1976)* Stewart (1976b)* Wood (1972)
Information load	Information load is a measure of the degree to which the interaction absorbs the memory and/or processing resources of the user.	In most tasks, user performance is adversely affected by information loads which are either too high or too low. Obviously, information load is a function of the task and the user's familiarity with the task, but is is also affected by the design of the dialogue. The information load associated with a task can be measured empirically or estimated, and can be altered by changing display mode and/or channel capacity, power of commands, use of default values, command language type and structure, and a variety of other dialogue details.	Finkleman (1976) Meister (1976) Parsons (1972) Treu (1975)
System response time		See next two figures.	
Communication medium		See later sections dealing with natural-language dialogue, interactive graphics, input device selection and output device selection.	

Figure 12 (Concluded).

parameter values, etc., in <u>response</u> to some form of prompting by the computer, such as a question or a list of alternatives. In user-initiated dialogue, no such prompting occurs, and the user must generate the necessary input from memory. "Mixed-initiative" dialogues, in which the dialogue may freely be "led" by either the user or the computer, also occur (e.g., Carbonell & Collins, 1970), but these tend to be limited to fairly sophisticated systems using natural language (see the later section on natural-language dialogue). Variable initiative, in which the user (or occasionally the system) selects either user- or computer-initiated dialogue, is much more common.

Computer-initiated dialogue has several advantages. It allows reliance on the passive vocabulary of the user (the set of words which the user can recognize and understand), which is typically much larger than the user's active vocabulary (words which the user can generate and use without prompting). It also allows the designer to implicitly convey to the user a "mental model" of the system's dialogue structure. This can be very helpful to the new or casual user, and may help to reduce the adverse effects of a satisficing strategy (Eason, 1976) by constantly reminding the user of the existence of unused commands. Computer-initiated dialogue can also reduce the clerical workload of the user by allowing selection, rather than typing, of words, and can even allow the selection of nonverbal command arguments in graphical systems. Common types of computer-initiated dialogue include menu selection, form-filling, and question-and-answer techniques, all of which will be discussed in the later section on dialogue types. Examples and discussion of particularly effective computer-initiated dialogue can be found in Thompson (1969, 1971) and Johnson (1977).

Computer-initiated dialogue also has two principal disadvantages. First, such a dialogue can constrain the amount of information conveyed in a single transaction, sometimes with detrimental results. For example, McAllister and Bell (1971) discuss a bibliographic system in which the user potentially goes through 84 separate frames just to order a book. While this may be an appropriate dialogue for the extreme novice,

it is almost certainly detrimental for the experienced user. Second, the use of computer-initiated dialogue can result in system-response-time delays which are unacceptable, especially for experienced users. This is a function not only of dialogue design, but also of hardware and software configuration. For example, computer-initiated light-pen, menu-selection dialogues for command construction may be extremely effective when controlled by a "smart" terminal -- whose response time may be psychologically negligible -- and may be totally disruptive when the same dialogue is controlled by a host computer with response times of a few seconds. A later section on response time treats this issue in more detail.

It is important to note that even the use of a rapid, "smart" terminal does not, by itself, resolve the first difficulty, which is a question of dialogue structure and transaction content. In general, though, actions which the user must frequently make must either be user-initiated or be part of a relatively rapid interchange, if the dialogue is to be satisfactory to the frequent user. If a computer-initiated dialogue would cause the response-time guidelines (discussed in the later section on response time) to be exceeded frequently by the expected system user, then user-initiated dialogue should be provided. Often, the best solution is a computer-initiated default dialogue, with a provision for the experienced user to override this mode. Nickerson and Pew (1971) present several good suggestions with respect to such dual-mode dialogues. For systems intended for dedicated, well trained users, a purely user-initiated dialogue may be satisfactory.

Flexibility, Complexity, and Power

Flexibility, complexity, and power are related, but clearly differentiable, dialogue properties. There is an understandable tendency on the part of designers to treat these properties quite simplistically: flexibility is good, complexity is bad, power is good. The limited body of empirical data bearing on these issues suggests

that this treatment is inappropriate. It is likely that each of these properties is related to performance in a nonmonotonic manner, and that the performance effects also vary with task and, especially, user properties.

Flexibility (as defined in the table) is difficult to measure in an existing dialogue design. It is possible, though, to devise dialogues which accomplish the same purpose, but differ primarily on this variable. Walther (1973; see also Walther & O'Neil, 1974) varied dialogue flexibility in a study involving varying levels of user experience. Highly flexible dialogues were found to help very experienced users, but to degrade performance of moderately experienced computer users to a significant degree, especially by increasing error rates. There is also evidence, from the MICA study (Eason, 1976; Stewart, 1976b), that users adopt a satisficing strategy with respect to utilization of flexible dialogues, and may thus fail to take advantage of the flexibility provided.

"Complexity," as the term is used by Carlisle (1974) is a measure of the number of alternatives from which the user must choose at any given point in the dialogue. This is not the same as structural complexity; in fact, "complexity" can be reduced by imposing a tree structure on the dialogue, which reduces the number of alternatives at any point, but increases the complexity of the dialogue structure. Carlisle studied the effect of interface "complexity" on users differing in experience and on several cognitive attributes (e.g., verbal intelligence). Although medium complexity (12 options at each node, as contrasted with 3 and 21 in the other conditions) yielded the most interaction and the greatest variety of command usage, performance effects were inconsistent. Complexity interacted somewhat with user properties.

Using nonsense syllables as "commands," Baker and Goldstein (1966) demonstrated that performance suffers when the displayed list of alternatives includes options not currently relevant. However, it is not

clear that this result is fully generalizable to dialogue involving meaningful commands and structure. In an unpublished study, Boies (1977) compared a language with a small number of generic commands and a large number of arguments against a language with many specific commands and few arguments. The generic organization was reportedly more useful.

Complexity, then, is clearly a relevant variable, but no clear guidance with respect to optimal complexity emerges from the surveyed literature. In passing, it might be noted that some designers are uncritically using Miller's (1962) "magical number seven plus or minus two" as a guideline for the optimal number of dialogue alternatives. It is not clear that short-term memory capacity is the limiting factor here, and this convention should be considered arbitrary.

Dialogue "power" is concerned with the amount of work done by one command. Where the user executes high-level procedures, each of which can be specified as a sequence of lower-level procedures, the designer has some choice of command level. For example, the user of a mathematical system might accomplish a matrix addition by executing a series of scalar additions, or a single matrix addition command might be provided (as in Wood, 1972). Clearly, the Boies study, discussed above, bears on this issue also. There is no standard metric of dialogue power, but Halstead's (1973) approach to the measurement of programming language power may be applicable within restricted task domains which have a fairly well accepted set of "primitive" procedural operations. In interactive dialogues, high power is usually accompanied by either high complexity or restricted generality. In the MICA survey, the latter was a significant factor in system rejection, especially by scientific and technical users (Stewart, 1976b).

Although no clear guidelines emerge with respect to these three dimensions, it is demonstrable that the prevalent attitude of the design community toward them is simplistic. A discussion of the issues in a

design guide may, at the very least, cause a more situation-specific consideration of these factors by the designer.

Information Load

The information load imposed on the user is a major factor in the success of many interactive systems. Although several authors have expressed a general concern about this problem (especially Finkelman, 1976; Treu, 1975), there is little evidence that existing knowledge of the measurement and effects of information load is being applied to computer system design. User performance can be adversely affected by information loads which are either too high (the usual problem) or too low. The performance effects of information load differ significantly depending on whether it is the memory or processing resources (or both) which are overloaded (Norman & Bobrow, 1975). This, in turn, is strongly affected by the complexity of the task and the experience of the user.

While the demands of the underlying task and the skill of the user are significant determinants of information load, there are many parameters of the interactive dialogue which can be used by the designer to ensure an acceptable information load. Among the major possibilities for reducing information load are:

- The use of display devices with higher channel capacity (e.g., graphics instead of alphanumerics)
- Reformatting displays for improved correspondence to the immediate information requirement of the user
- Use of commands of appropriate power for the immediate task of the user
- Use of a language which minimizes information processing by the user (e.g., natural language, a command language with appropriate complexity)
- Moving clerical operations (e.g., manipulations of data before data entry) into the system

Structuring command language to correspond to the substructure of the user's task

Use of default values.

The more basic issue is the fact that user information load is often not a formal consideration during system design. It is possible to measure user information load during actual or simulated system use, by the use of competing tasks, for example (see Norman & Bobrow, 1975), and information load can be estimated from detailed task-analytic data. Even very gross task analytic data can be used to obtain a rough estimate. A design guide should review the effects of information load on user performance. This information is found largely in non-computer-related material (e.g., Meister, 1976). The design guide should suggest procedures for determination of information load, including both precise and rough measures. Finally, the design factors which can be used to influence information load should be discussed.

System Response Time

System response time is an area in which human factors considerations are clearly of concern to the system designer. There is an obvious tradeoff here between performance (slow system response reduces user satisfaction and degrades user performance) and cost (rapid response can increase costs dramatically). In some quarters, there is a tendency to dismiss this issue as moot because of rapid advances in technology which allow ever shorter response delay at reduced cost. It should be kept in mind, however, that the shift to networking, distributed processing, front-end processors, etc., presents a whole new pattern of response delays due to communication time, shared processing, and so on. Human factors issues associated with system response can easily affect the basic architecture of these systems, and such considerations remain a significant element in cost-performance tradeoffs even in conventional systems.

The human factors concern here is with the effects, on both user performance and user satisfaction, of delays in system response and of variations in those delays. These effects depend in part upon the user's expectations with respect to system performance, and the user's impression of the system activity required to process a command or query (Carbonell et al, 1968). The user's mental model of the system may not correspond to the designer's or to reality, so that the expectancies involved here may have implications for training. Most importantly, the effects of response delays depend upon the relationship of the particular transaction to the user's ongoing problem-solving process (Miller, 1968).

There are several related components of response delay, as illustrated and defined in Figure 13. Notice that, for the purpose of this discussion, SRT is defined as the interval from the user inquiry to completion of the resulting display, including writing time. A summary discussion of the effects of response delays and SRT variation is presented in Figure 14.

The disruptive effect of excessive SRT on user performance appears to result largely from interference with the user's "continuity of thought" (Miller, 1968). This is a highly relative concept, involving very short intervals of time (seconds or fractions of seconds) for procedural steps in the construction of a command, and perhaps quite long intervals (many minutes) when the user has reached "closure" with respect to a basic problem-solving step. In the former case, the disruption is probably related to human short-term memory limits and attentional phenomena. The key to satisfactory performance here may be maintenance of a "conversational" interaction. In this instance, the key interval may be SRIT, rather than SRT.

In some situations, SRT variation may be a more important human factors problem than long delays. Several authors have speculated that such variation can be demotivating to the user, resulting in both

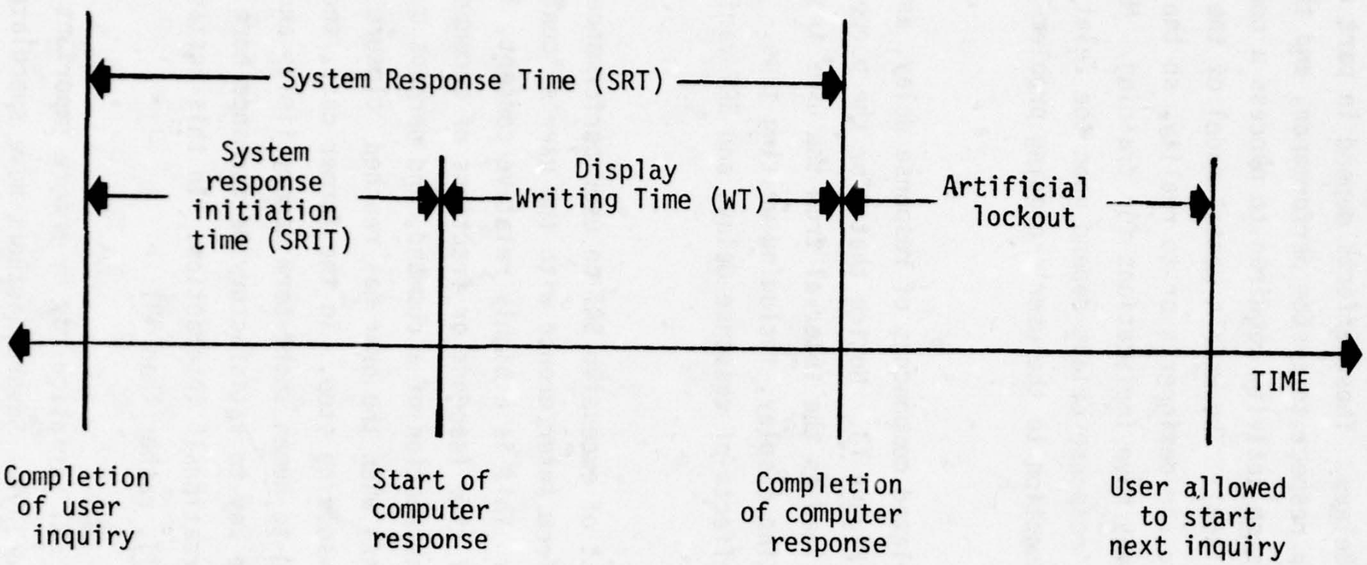


Figure 13. Definition of System Response Time and Related System Temporal Properties.

Factor	Comments	Principal References
System response time	If below user preparation time, involves little or no performance advantage, and may, in certain cases, actually degrade user performance	Franklin & Dean (1974)* Newman (1969)**
	Otherwise, delay in excess of that tolerable for specific task results in step decrements in performance and results in user dissatisfaction.	Miller (1968) Morefield et al (1969)**
	With excessive delay, user may adapt mode of system use by time-sharing own time or by a variety of partial or total withdrawal strategies	Carbonell et al (1968) Eason (1976)
System response time variation	If significant, may result in user dissatisfaction. Performance effects are probably mostly related to individual SRTs, as discussed above.	Carbonell et al (1968) Simon (1966) Williams (1975)**
Display writing time	Has been subjected to limited study as separate issue. No clear conclusions.	
Artificial lockout	May improve performance in complex problem solving. Reduces user satisfaction.	Boehm et al (1971)** Gold (1967) Seven et al (1971)**

Figure 14. Summary of System Response Time Effects

performance decrements and user dissatisfaction. Williams' (1975) data do not support this speculation, but do not persuade us to abandon it, either. More research is needed to justify a firm conclusion.

SRT guidelines can be offered which appear quite reasonable from a psychological viewpoint, although little explicit empirical verification can be claimed. Miller (1968) presented SRT guidelines for 17 situations ranging from single key depressions to completion of a request for a major computation. Engel and Granda (1975) have altered these slightly, and have also presented SRT guidelines (attributed to W. C. Allison and R. F. Challman) in terms of the user's perception of the system activity rquired to satisfy his inquiry. In the context of an extensive design guide, it should be possible to relate such guidelines more clearly to user tasks. This information might be supplemented with data on user preparation times (inter-response intervals) associated with procedural steps in command construction and other "conversational" transactions. Several of the papers surveyed present data from which rough estimates might be derived for various dialogue operations at this level. The resulting guidelines can be related more clearly to specific design decisions, such as terminal selection and distributed processing system architecture.

"Artificial lockout" is a separate, but related issue. This involves restricting a problem solver's access to the computer for some period of time after presentation of results from the current request. There appears to be some evidence that this may cause the user (in a complex problem-solving situation) to concentrate more on the problem to be solved and less on the tactics used to solve it, with resulting improved performance, at some cost in user satisfaction. This technique is related to Stewart's (1976b) advocacy of dialogue techniques intended to break the user's problem-solving "set". Although a design guide could make the designer aware of this technique and its possible advantages and disadvantages, it is not clear that more detailed guidance can be offered.

TYPES OF INTERACTIVE DIALOGUE

The selection of a dialogue type or types is one of the most important decisions made in the course of designing an interactive system. This section will discuss several major dialogue types and outline what is known about their advantages, disadvantages, and detailed design, and about the potential for development of human factors guidelines for each type. In addition to the major dialogue types discussed here, there are many minor variants and many possibilities for combining dialogue types. Although space does not permit a discussion of all of these here, a detailed design guide must recognize their existence and probably provide assistance with particularly common combinations (e.g., menu selection or user-initiated mnemonic commands combined with form-filling for data entry).

A major controversy involves the applicability of a human-to-human communication model to person-computer dialogue. The proponents of such a model argue that interactive dialogue which operates in the same way that humans communicate with one another is inherently natural and flexible, requires little user training, and is very powerful. The extreme advocates of this view conclude that all computer systems should communicate in natural language (e.g., English, preferably spoken), perhaps supplemented with interactive graphics.

The opponents of this view usually agree that "natural language" is natural for the user, flexible, etc. They point out, though, that it is also highly ambiguous, and that a great deal of general and problem-specific knowledge is required to comprehend interpersonal dialogue. They argue that natural-language dialogue will not be cost-effective, except in rare instances, for many years, and that, in most applications, it may even be an inherently poor substitute for simple, umambiguous command languages, menu selection dialogues, etc.

It seems appropriate to decompose this controversy into three different components, which often seem to be confused with one another:

(1) natural-language dialogue, (2) "natural" dialogue, and (3) semantic knowledge.

The question of "natural-language" (e.g., English) dialogue will be discussed in a later section, but warrants a brief treatment here. We are now able to achieve person-computer interaction via "natural language", provided the domain of conversation is limited, and the "natural language" is itself restricted in syntax and vocabulary. Furthermore, the language must usually be "formal" natural language, such as one might find in a textbook on (for example) English grammar. Unfortunately, we are also beginning to understand that this isn't the way people actually talk to one another. Independently of the technological issues of automated production and "comprehension" of natural language, we need research like that of Chapanis (1975, 1976) and his colleagues to determine the nature of the <u>actual</u> language used in interpersonal dialogue and the tolerance of humans for various constraints on their language. And this research has a long way to go before it will be directly applicable to system design.

The second component of the controversy is the issue of "natural" dialogue. Dialogue need not be conducted in natural language to be "natural", in the sense that it requires little training, "feels right", etc. In some domains (e.g., mathematics, engineering drawing), natural language is clearly not the most appropriate dialogue type. It is likely, however, that much of what we know about the nonlingual aspects of natural conversation (continuity, response time, etc.) is applicable to the achievement of natural dialogue, even if the dialogue is conducted via a simple command language. This aspect of natural dialogue has been discussed very well by Foley and Wallace (1974).

The third component concerns semantic knowledge. Satisfactory natural-language processing requires that the computer have considerable semantic information about the application domain and about prior conversation in order to disambiguate, and ultimately to comprehend, the user's statements. Such knowledge enables the system to respond appropriately

in the context of the user's previous interactions with the comput[e]
adapting its interpretations and responses appropriately, more or l[ess]
as a human would. It does not follow, though, that such knowledge [is]
only relevant when natural-language techniques are used. Regardles[s]
of dialogue type, "knowledge-based systems" have great (but expens[ive]
and long-range) potential for adapting to the user's needs, style,
specific problem (see Waksman, 1974). This is the basis of most o[f]
proposals for "personalized" systems (e.g., Negroponte, 1975).

It seems likely that syntactic aspects of natural-language a[re]
not the most important of these three issues. In any case, it is t[he]
"natural dialogue" issue which can be most readily addressed by a d[esign]
guide in the near future. A design guide might be able to acquaint
designer with some of the advantages, disadvantages, and complexiti[es]
of natural-language processing and knowledge-based systems, but it
not attempt to educate the designer in the implementation of such s[ystems].
If designers wish to employ such techniques, and do not already hav[e]
knowledge to do so, they would be well advised to contact a special[ist].

The remainder of this section is divided into subsections de[aling]
with specific dialogue types. Thus, a discussion of command langua[ge]
properties appears in the section dealing with "user-initiated comm[and]
languages", even though some of the material is potentially relevan[t]
the design of command languages for menu-selection dialogues. This
zation seems simpler, for the current report, than the more highly
tioned organization which a design guide would need. It does, howe[ver],
require the reader to recognize that the discussions of language pr[o]-
perties are more broadly applicable than their placement in the rep[ort]
might suggest.

There is a potential for the construction of guidelines whic[h]
address dialogue details, but are relatively independent of dialogu[e]
type. Potential topics for these guidelines include such things as
deletion by a two-step process, "undelete" capability, error messag[es]

features, provision of system status information, feedback and reinforcement of the user, and command stacking. Good, though brief, guidelines of this sort already exist (especially in Engel & Granda, 1975; Nickerson & Pew, 1971), but need to be made more comprehensive if the design guide is to cover a wide range of dialogue types.

We have attempted to distinguish dialogue types purely on the basis of dialogue properties, without regard to the purpose for which the dialogue is intended. Thus, we address such dialogue types as form-filling, menu selection, and natural language (see Figure 15). Martin's (1973) excellent book on interactive dialogue contains a much more extensive taxonomy of this sort. Others (e.g., Miller & Thomas, 1977; Nicholson et al, 1972) have found it useful to classify dialogues primarily by purpose. For example, Miller and Thomas discuss the functions which should be provided in specific types of interactive software, such as editors, file managers, query systems, programming aids, and message processing facilities. Discussions of this sort could provide a very useful source of information for design guidelines intended for a restricted class of systems, but are less clearly useable for a general-purpose design guide. Statistical studies of command usage (see Figure 4) may also be more useful when dealing with specific application areas. Examples of effective dialogue (e.g., Borko, 1966) are easier to present in such a context, but Martin (1973) has done an effective job with examples in his general treatise.

Question-and-Answer

In a question-and-answer (Q & A) dialogue, the user responds to questions asked by the computer. For the computer-naive user, this is probably the simplest dialogue type. Card et al (1974) and Lucas (1977) report a highly successful application of this dialogue type in medical interviewing of totally naive users. Only yes-no questions were used, and the user had a keyboard with three keys marked "Yes", "No", and "?". With no training, users successfully completed the questionnaires and reported satisfaction with the system.

Dialogue Type	Description	Comments	Principal References
Question-and-Answer	Computer asks a series of questions, to which user responds.	Inherently computer-initiated. For totally naive user, this is probably the most error-free dialogue type. This approach rapidly becomes cumbersome as the user gains experience.	Card et al (1974)** Lucas (1977)**
Form-filling	Computer presents form with blanks. User fills in blanks.	Computer-initiated. Faster than ordinary question-and-answer dialogue, because user provides several responses in a single transaction. When user input is dominated by parameter values, rather than commands, this approach is often best. Other than casual use requires terminal with tabbing feature. Under some circumstances, a significant proportion of syntactic data entry errors may be detected if terminal has provision for imposing constraints on data by field.	Martin (1973) Pew & Rollins (1975) Strub (1975)**
Menu Selection	Computer presents list of alternatives, and user selects one or more.	Inherently computer-initiated. Can be used for command construction as well as data base search. Very natural dialogue if response-time criteria are satisfied and "point-in" selection device (e.g., light pen, touch panel) is used.	Martin (1973) Ridsdale (1970)* Thompson (1969, 1971) Uber et al (1968)
Function Keys with Command Language	User indicates desired action by depressing keys, each of which represents a command, command modifier, or parameter value.	User-initiated, but with keyboard as memory aid. Can be computer-initiated if "programmable" keyboard or tutorial displays are used. Often appropriate when user input is dominated by commands, rather than parameter values. Appropriate for naive user only if command syntax is very simple and/or computer-initiated form is used; otherwise requires training. Constant presence of all commands and modifiers may make it difficult for user to learn appropriate model (e.g., hierarchical structure) of command language, if language is not simple.	Martin (1973)
User-initiated Command Language	User types commands, perhaps using mnemonic abbreviations.	Acceptable approach for well-trained user who has fully internalized model of system function and language syntax. Otherwise error-prone and sometimes frustrating. Usually preferred by system designers and programmers, who tend to satisfy these criteria. Often applied uncritically by them to systems in which user does not satisfy criteria.	Martin (1973)
Query Language	User inputs questions or data-base access procedures to a data base system. System produces response or report.	Several existing query languages appear to be useable by both novices and programmers, but many errors occur in their use. The problem areas appear to be reasonably well known, but only fairly general solutions can be confidently prescribed. Detailed guidelines for query language design would be premature.	Gould & Ascher (1975)** Reisner (1977)** Thomas (1976)* Thomas & Gould (1975)**
Natural Language	Dialogue is conducted in user's natural language (e.g., English).	Can be user- or computer-initiated or mixed initiative. Fairly high-powered natural language capabilities are now achievable. Cost is very high, however, since the system requires an extensive "knowledge" of the application area in order to understand user inputs. Development of such a data base is definitely a nontrivial task, and is not for unsophisticated designers. "Natural" language may not be the most "natural" dialogue type for many applications (e.g., engineering drawing, mathematics).	See text discussion
Interactive Graphics	Generation of pictorial displays, ability of user to select displayed entities and spatial locations by pointing or similar nonverbal means.	This is not truly a dialogue type, but interactive graphics offers great additional dialogue flexibility. With rapid response (as by use of smart terminal with refreshed graphical CRT display), very rapid, flexible and "natural" dialogues are possible. Interactive graphics is relatively expensive, but costs are dropping. Performance improvements may offset extra cost even in some relatively mundane applications (e.g., search of hierarchical data base). Most research on interactive graphics is concerned with details of input and output devices and techniques, rather than overall dialogue properties. See later sections on those topics.	Foley & Wallace (1974) Martin (1973)

Figure 15. Basic Interactive Dialogue Types.

Q & A dialogues can be used for much more sophisticated situations, and are often used for data entry tasks. Experience suggests that this dialogue type is unsatisfactory for systems with constant or highly experienced users, but no data bearing on this issue are known to exist.

Form-Filling

Form-filling is often used in situations in which the user's input is dominated by parameter values, rather than commands. This dialogue mode naturally allows multiple user responses in a single transaction and may provide more contextual information concerning the needed information than does a Q & A dialogue. Effective use requires a CRT terminal or similar device in which user input can be positioned on a two-dimensional display, and requires a terminal tabbing feature.

Martin (1973) illustrates several versions of form-filling dialogues. As with Q & A dialogues, little empirical data exist. Strub (1975) compared form-filling with unaided user-initiated input of text data into a message-handling system. Unaided input was 40% slower, but this is not a statistically reliable result because of the study's small sample size.

Pew and Rollins (1975) present a number of useful recommendations with respect to the detailed design of a form-filling dialogue. This information is intended for use by system designers and is in a form appropriate for incorporation into a design guide. It is not highly detailed, however, and should probably be extended somewhat.

Menu Selection

Menu selection is the archetype of computer-initiated dialogue. The item(s) to be selected appears directly on the display. The user need only recognize the desired action, whereas a form-filling or Q & A

dialogue may only prompt the user for a response which must still be remembered. Furthermore, a simple menu-selection dialogue ordinarily requires only one user action per choice, rather than, for example, the series of keystrokes required to type a whole word.

Martin (1973) provides examples of several types of menu-selection dialogue. Once again, there is little empirical data pertaining to the circumstances under which this particular dialogue type is appropriate, although Ridsdale (1970) reports an empirical study of a menu selection dialogue used effectively by hospital personnel for medical data entry. Among the study's conclusions are that the "branching questionnaire" technique was effective for this data entry task, and that menu selection is an appropriate form of dialogue for novice users.

Because of its reliance on the user's passive vocabulary and recognition memory, menu selection is a particularly natural dialogue method for hierarchic search. It is, therefore, very popular for a variety of information retrieval dialogues. Thompson (1969, 1971) provides an excellent discussion of such dialogues, with examples and some guidelines for the designer. Uber et al (1968) discuss this and other applications of menu selection, and provide well-considered suggestions for such properties of the dialogue as list size and display format.

Menu selection is also useful for construction of commands, when a single command has several elements such as a verb and a series of arguments. The literature contains less information about this application, probably because of the response-time problems it creates. When several user actions are required to support the construction of a single command, it is usually necessary that any system response which must occur within this process be very rapid (e.g., a few hundred milliseconds). Thus, menu selection is ordinarily not an appropriate dialogue type for command construction unless smart terminals or a dedicated host computer are used. When response _is_ rapid, however, this dialogue type is very effective for novices and experienced users alike.

Except for users who are skilled typists, menu selection is most effective when a "point-in" input device (e.g., light pen, touch panel) is used (see Earl & Goff, 1965). When an ordinary alphanumeric terminal is used, it is usually necessary to number the items, or provide some similar coding scheme, so that the user can identify the desired option by code. Pew and Rollins (1975) provide some reasonable suggestions for designers using this approach. Mode mixing is an important problem when a point-in selection device is used. If the user must switch back and forth between option selection (especially with a light pen) and typing, performance may be adversely affected (Earl & Goff, 1965). Under these circumstances, the designer should consider either (1) an extraordinary effort to place all information on the display in selectable form, (2) use of a point-in device which minimizes mode mixing problems (e.g., a trackball), or (3) use of typed codes to select menu options.

There are a number of scattered studies which provide information useful in a design-guide discussion of menu selection. For example, Baker and Goldstein's (1966) finding that display of potentially-but-not-currently-relevant choices interfered with user performance is relevant. Ramsey (unpublished study) found that serial presentation of successive menus in the same area of the display is preferable to simultaneous presentation in different areas. Engel and Granda (1975) provide useful guidelines for formatting of menu selection displays. In general, the information in the literature on this topic has not been adequately integrated, but could be, if a design guide is undertaken.

Function Keys With Command Language

Dialogues which use function keys typically have some of the properties of computer-initiated dialogue and some user-initiated properties. Although the user may have to construct a meaningful sequence of key depressions without assistance from the computer, the

keyboard itself provides a memory aid which allows the user to rely on recognition, rather than recall, memory. If "programmable" function keys, with back-lighting or even variable legends, are used, the dialogue can be a pure menu-selection dialogue. Ordinarily, though, all options are present on the keyboard, including those not currently relevant.

Although many systems use function keys, and there is a good deal of data available concerning the desirable properties of the keyboard itself, there is very little information in the literature concerning the dialogue-related aspects of function-key interfaces. The best presentation of this topic is in Martin (1973), although some discussions of specific systems (e.g., Wood, 1972) contain useful information on function-key dialogues.

User-Initiated Command Language

The most common dialogue type appears to be the user-initiated command language. This dialogue method is the easiest and most efficient from the computer's viewpoint, involves the lowest electronic communication overhead, minimizes user waiting time during command construction, and is the most compatible with ordinary teletypewriter terminals. It is generally preferred by system designers and programmers for these reasons, and perhaps because it is more like ordinary computer programming than the other dialogue modes.

This is also the dialogue mode which inherently provides the least assistance to the user in acquiring a "mental model" of the system and a knowledge of the functions and syntax of the language. Furthermore, the syntax tends to be more involved, requiring separators (commas, parentheses, etc.) for unambiguous lexical analysis and parsing. Except for very simple command languages, this dialogue mode requires a trained user, with the amount of training varying directly with the complexity of the language.

A user-initiated command language is probably satisfactory for computer-sophisticated users and more or less dedicated users who can be expected to undergo significant training. In other contexts, this dialogue method is used more frequently than it should be, and has been a significant source of errors and system rejection by relatively unsophisticated users.

The above assertions are the opinions of the authors. Although there have been a few empirical studies dealing with the desirable properties of command languages, we found no empirical data evaluating this basic approach to interactive dialogue, or comparing it with others.

We also found no well-integrated source of information to assist in the design of such a command language, although the information which was found might support such an integrated treatment. Much of the basic research which deals with "command languages" is concerned with the ability of humans to comprehend assertions, or to comprehend and execute commands, rather than their ability to generate and/or recall commands in order to accomplish their own purposes. Several such studies are reviewed by J. Newman (1976). Although this research may eventually contribute to our understanding of command languages, these two tasks are sufficiently different that any attempt to generalize directly from one to the other should be viewed with considerable suspicion.

Similarly, most of the research which has been done on programming language properties is inapplicable to command language design (although several of the studies provide insight into query language design, as discussed in the next section). Most of the studies deal with language properties not usually relevant to command languages (e.g., transfer of control, indentation), and almost all use programmers as subjects.

What remains are a handful of studies bearing on specific properties of command languages, and several much more general discussions. Watson (1976) provides a good discussion of some of the issues in command

language design, presents examples from the command language of the NLS system, and discusses the reasoning behind the selection of its properties. Although Foley and Wallace (1974) emphasize interactive graphics, their excellent discussion of general language properties is relevant to even simple command languages. Sayani (1976) presents a reasonably good discussion of command languages for operating systems, with an emphasis on matching the language to the user's "mental model" of the system. Treu (1975) discusses this correspondence in more detail, and provides a model of the sources of "mental work" involved in mapping between the user's model of actions to be performed and the set of commands available to perform them. While it is difficult to see how the model might be tested or applied, it may help in thinking about the problem.

With respect to overall command language structure, the studies by Carlisle (1974) and Boies (1977), discussed previously in the section on "complexity" are suggestive, but by no means definitive.

A topic of greater research focus is the use of positional or keyword syntax for parameter values in commands. Commands with keyword parameters typically have a form such as

COMMAND(KEYWORD=VALUE, KEYWORD=VALUE)

while positional parameters are identified by the position of the values in a list, as

COMMAND(VALUE,VALUE).

Although the positional syntax is more concise, it involves higher memory loads for the user, because of the implicit information which the user must provide. Heafner (1975) presents indirect evidence that programmer users of a command language prefer a keyword syntax. Weinberg (1971) mentions (but does not describe) a study in which programmers, working with relatively unfamiliar material, performed faster with a positional syntax, but made 2-4 times as many errors as with a keyword syntax.

Abbreviation of commands and keywords is also a relevant topic, on which little specific research has been done. Watson (1976) describes a system in which several command abbreviation schemes are available to the user. Of these, the most frequently used consists of single-character abbreviations for the more commonly used commands (less common commands with conflicting first characters require longer abbreviations for disambiguation). "First-k-character" algorithms are frequently used. Such abbreviated input is important for the experienced, frequent user, and should be consistent with unabbreviated command usage in order to provide a smooth transition for the user-in-training.

The use of default values is also helpful in promoting natural and concise dialogue. Worthwhile suggestions are found in Gilb and Weinberg (1977) and Martin (1973), but no directly applicable research was noted.

Existing "guidelines" papers (e.g., Engel & Granda, 1975; Nickerson & Pew, 1971) provide little specific help with command language design. They do provide general suggestions concerning such matters as error messaging and overall style. There do not appear to be sufficient research data to support definitive guidelines in this area, but this is an area in which general knowledge of human information processing and basic human factors principles could provide considerable guidance. Without being overly speculative, it should be possible to develop reasonable human factors guidelines dealing with command language design. It is somewhat surprising that this topic has received so little attention on the human factors side of the literature. Guidelines in this area should address such topics as command language structure and complexity; statement syntax, including keyword and positional forms; separators and terminators; abbreviations and processing thereof; default values; command choice, including minimization of both syntactic and semantic confusability; mechanisms for switching to computer-initiated form; error messaging and recovery; command stacking; and overall dialogue style.

Query Languages

In a query language, the user interacts with a system which has access to a data base. The query language is used to express the user's specific request(s) for factual information. Ordinarily, query languages do not alter the data base, but merely allow the user to ask questions about the data.

Research in this area has been mainly of two types. Several very basic research efforts have allowed computer-naive personnel to express queries in natural language or in more constrained languages, and have attempted to characterize their behavior in order to provide insight into query language use. Other studies have investigated the useability of specific available query languages, usually not in a comparative fashion. In most studies, the experimenter has provided queries which were to be translated into a specified language. The query language user, however, starts with a need for information, and must formulate a question, plan an approach, and encode the query. Only Gould and Ascher (1975) have really addressed this overall process.

The studies of specific languages have generally concluded that existing query languages can be learned and used by both novice and computer-sophisticated users. The languages investigated include a variant of IBM's Interactive Query Facility (Gould & Ascher, 1975), which had relatively high error rates, Zloof's "Query by Example" (Thomas & Gould, 1975) and SQUARE and SEQUEL (Reisner, 1977; Reisner et al 1975). Although these languages differ considerably in basic philosophy and in details of the dialogue, only one study (Reisner) has attempted any sort of comparison, and that involved two relatively similar languages. Essentially, comparative data on basic approaches to query languages appear to be nonexistant.

Although the languages listed above were basically useable, many errors were made in their use, and many errors emerged from the more

basic studies which did not deal with currently automated languages. These errors provide considerable insight into the kinds of logical constructions and query language features which present difficulties for users (see Figure 16). All of the results appear to be basically consistent with older research on concept identification and related topics.

In some cases, fairly specific conclusions concerning language features seem warranted. For example, a user probably should not be required to convert "over 50" to "51 or more", and elimination of the word "or" from query languages might avoid errors (Gould & Ascher recommend alternatives involving special punctuation and an "ALSO" construct). At a somewhat higher level, Reisner (1977) concludes from her experiment, on both novice and programmer users, that a "layered" or partitioned query language design is appropriate and makes suggestions for its properties. For the most part, however, insufficient data exist to support broad, detailed guidelines concerning query language design.

In addition to a number of specific, but scattered recommendations such as those above, some higher-level approaches have been suggested which may make query languages easier to use and less error-prone. The most broadly applicable of these is a restatement of the user's query, by the system, before execution (Gould & Ascher, 1975). Such a restatement should paraphrase the user's input, rather than repeating it as entered, and should be suppressible by the user (it should probably be included with printed output even if suppressed. Clearly, many of the categories of error described in Figure 16 might be detected by an attentive user if such feedback were provided.

Several investigators have recommended more active aids to query formulation. Typically, these aids have been proposed in a restricted information retrieval domain (usually bibliographic search) and have involved a gradual interactive negotiation between the system and the user (e.g., Caruso, 1970; Green 1967; Summit, 1971). In some

Problem Area	Comments	Principal References
Logical Quantifiers	Use of logical quantifiers (all, some, none) in the presence of set relations (union, intersection, etc.) is very error-prone.	Thomas (1976)**
Set Relations	When sets of elements are related in a complex way, human interpretation of the relationship is often erroneous.	Thomas (1976)**
Logical Relations	Disjunction (logical "or") and negation are error-prone constructs. Although this finding is consistent with basic psychological research and is probably generally true, the principal study (Miller, 1974) in which it is related to programming and query language design involves an atypical task in which the subject must compensate for absence of these constructs in a language by devising a procedural specification using transfer of control.	Gould & Ascher (1975)**
Arithmetic Relations	Conversion of inequalities (e.g., from "over 50" years old to "51 or more" years old) is error-prone. Also, users tend to use arithmetic relations even with "nominal" categories, as "college degree greater than or equal to B.S.".	Gould & Ascher (1975)**
Semantic Confusion of Commands	Errors occur when query language has confusable commands, such as COUNT ("how many numbers are there?") and TOTAL ("what is their sum?").	Gould & Ascher (1975)**
Use of Synonyms for File Names, Properties, etc.	Users tend to substitute synonymous terms (e.g., "employee" for "personnel" file) which system may not recognize.	Reisner (1977)**
Misspelling	Spelling errors are common in query formulation. Also, users tend to use an incorrect ending (e.g., "employees" instead of "employee").	Gould & Ascher (1975)** Reisner (1977)** Thomas & Gould (1975)**
Omission of Problem-Relevant Attributes	In formulating complex queries, users frequently omit one or more of the attributes which define the set.	Gould & Ascher (1975)** Thomas & Gould (1975)**
Contextual Referencing	If unconstrained, users tend to make contextual references in queries. However, there is no clear evidence that users fail to adapt to query languages which preclude such references.	Miller & Becker (1974)**

Figure 16. Some Known Problem Areas in the Use of Query Languages.

cases, a serial partitioning of the task is advocated. For example, Wilde (1969) deals with the difficulty of set relations by partitioning the query formulation into a search term identification phase and a second phase in which the logical relations among the terms are specified. Clarke (1970) suggests aids which even assist the user in determining appropriate relations.

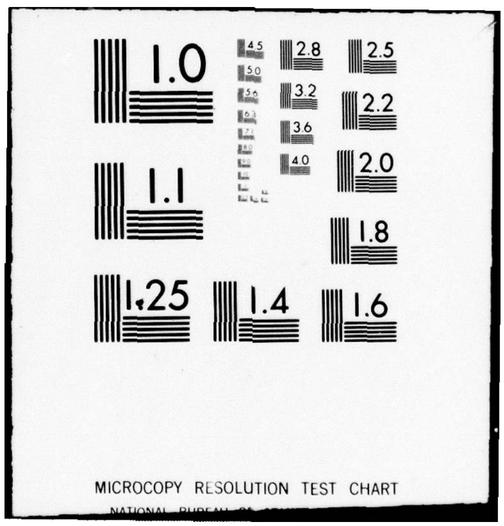
An issue which has received particular attention is the ability of users to specify queries in procedural form. Computer programmers typically express queries in the form of explicit procedures which, if executed, will presumably lead to the desired answer. Miller and Becker (1974) found that unconstrained naive users tend to use nonprocedural, or descriptive, specifications which are often ambiguous or incomplete, make contextual references, and refer to data aggregates rather than individual data elements. However, a later study by Gould et al (1976) seems to suggest that this need not be a problem. They found that, with proper task instructions, naive users could employ a restricted-syntax procedural query language. In fact, the users voluntarily continued to use this language, in preference to natural language, once they had been exposed to it. Highly procedural query languages are only one step removed from programming languages, and there is a great deal of literature on their design (some of it based on empirical study). The survey did not include literature on this topic, however.

An excellent study by Durding et al (1974) identified some capabilities and limitations of users in dealing with data bases having various structures (e.g., network, list, table, hierarchy). The basic finding is that users can work with all of these structures, but only when they match the user's perception of the underlying structure of the problem. It seems to follow that data base systems and query languages which arbitrarily restrict the data organization to a particular type or types may result in user errors, such as those observed in the study, when those systems are applied to problems whose "natural" data structure is different.

AD-A075 679 SCIENCE APPLICATIONS INC ENGLEWOOD CO F/G 9/2
HUMAN FACTORS IN COMPUTER SYSTEMS: A REVIEW OF THE LITERATURE.(U)
SEP 79 H R RAMSEY , M E ATWOOD N00014-76-C-0866
UNCLASSIFIED SAI-79-111-DEN NL

2 OF 2
ADA 075679

END
DATE FILMED
12-79
DDC


1.0
1.1
1.25
1.4
1.6
1.8
2.0
2.2
2.5
2.8
3.2
3.6
4.0
4.5
5.0
MICROCOPY RESOLUTION TEST CHART

Natural-Language Dialogue

Natural-language dialogue is still a highly experimental area. Although many command languages exist which happen to use verbs and nouns taken from a natural language, these are not at all what is meant by "natural-language dialogue". The ultimate goal here is to be able to converse with the computer in the same way as with another person. This is a much more complicated issue, as was pointed out in the earlier discussion of "natural" vs "natural-language" dialogue.

Natural-language dialogue is intended to allow for a great deal of flexibility on the part of the user. The inherent flexibility of interpersonal dialogue, however, makes this a difficult goal to achieve. Human communication is characterized by ungrammatical utterances and syntactic and semantic ambiguities. Several authors (e.g., Hill, 1972; Montgomery, 1972) have cited these difficulties and have argued that these difficulties prohibit the development of natural-language dialogues.

It is difficult to describe the current state of the art in this area because of its rapidly changing characteristics. There is a great deal of research effort being expended here, and frequent reports of "success". Yet the best results to date are systems which are very sophisticated and expensive to use, require extensive "knowledge" of the task domain, and have somewhat limited vocabularies. Since operational use of unconstrained natural-language dialogue is not yet feasible, the dialogue must be constrained in some way. The current literature on man-computer interaction primarily considers two such constraints and their effects on user performance. Either the vocabulary or the syntactic structures of the language are limited.

Syntactic Constraints and Disambiguation

There is some evidence (e.g., Carbonell, 1970) that limiting the syntactic structures that can be employed may not seriously affect the "naturalness" of the dialogue, depending on the number and nature

of the syntactic structures allowed. However, it is not clear what happens to the useability of an otherwise natural language as syntactic constraints are applied. Significant dialogue-processing difficulties can be eliminated by not allowing some of the more complex and subtle constructs of natural language (e.g., center embedding, contextual references).

It may be that elimination of such constructs would not have significant adverse effects on performance. On the other hand, a dialogue which seems natural may encourage the use of normal conversational language, including those constructs which have been eliminated. If it is unclear why the user's apparently clear inputs are rejected or even misinterpreted, the adverse effects on performance and system acceptance could be quite significant. Research is needed to investigate the effects on user performance of those types of syntactic constraints most likely in computer dialogue.

Another aspect of syntax which is particularly important concerns spelling and grammatical errors and ambiguities. Although such errors generally present little difficulty in processing interpersonal dialogue, they pose serious problems for the designer of an interactive system. The approach that is frequently taken is to assume that all user inputs are free of spelling and grammatical errors or ambiguities. In this case, the success of the user-computer dialogue depends, in large part, on the user's typing (or other input) and language skills.

If user inputs do contain ambiguities, it is desirable for the computer system, rather than the user, to detect the ambiguity and initiate some sort of disambiguation procedure. In the systems described by Codd (1974) and Plath (1972), for example, ambiguities are resolved by presenting the user with a list of possible interpretations and asking the user to select the appropriate one. These systems are designed for limited domains, however, and determining alternative interpretations is somewhat simplified by this fact.

Ideally, the system should be able to perform part, or all, of the disambiguation procedure with no input from the user. Although such a system has been implemented in the context of an interactive programming language (Teitelman, 1972) where the syntax and semantics are rigidly constrained, it is much more difficult to implement in a natural-language dialogue.

Vocabulary Constraints

The second major area in which natural-language dialogue systems commonly constrain the dialogue is in the semantics, or vocabulary, involved. In part, the vocabulary that the user would be expected to use is a function of the application area. It is only in extremely simple domains, however, that the meaning of any word, or group of words, can be clearly and unambiguously defined. In most areas of interest to interactive system designers, semantic complexity, in totally unconstrained dialogues, cannot be avoided.

As with syntactic ambiguities, a common method for circumventing these problems is to restrict the vocabulary that the user can employ. Some systems do not constrain, in an obvious manner, the semantic structures that the user can employ; rather, the system is able to process only a subset of the possible structures and "ignores" other structures.

The vocabulary that could be used in either interpersonal or user-computer dialogue can be viewed as consisting of two subsets. The first subset contains common, "general-purpose" words and the second consists of words that are generally relevant to a particular application area. For example, bibliographic search, command and control, etc., have rather specialized vocabularies. In a general sense, the application-specific vocabulary is used to convey essential information while the general-purpose vocabulary conveys little actual information but adds "naturalness" to the dialogue.

In problem-solving contexts, it has been demonstrated that vocabulary limitations on interpersonal dialogue do not seriously affect performance, provided that these limitations do not affect the problem-

specific words (Kelly, 1975; Ford 1977). A similar result would be expected in natural-language dialogue, provided that the vocabulary was selected to include the necessary words for an application area.

The systems described by Codd (1974) and Plath (1972) employ semantic restrictions, in addition to syntactic restrictions, and ambiguities are handled in the manner described earlier in the context of syntactic ambiguities. The system developed by Carbonell (1970) is a good example of a system that limits the vocabulary to a subset that is selected on the basis of the application area.

Comments

Progress toward eliminating these limitations and allowing for completely unconstrained dialogues is being made in the area of artificial intelligence. Although this approach is, ultimately, the most desirable, it is unlikely that all of the associated problems will be overcome in the near future. Examples of such systems can be found in J. S. Brown et al (1974), Carbonell and Collins (1970), and Grignetti et al (1974). Currently, such systems are limited to small, well-defined application areas and they can not be applied directly to other domains.

Eventually, the knowledge and techniques being generated by the development of such systems may be available to designers of interactive systems. It is unlikely, however, that any of these benefits will be available in the near future. It is also important to recognize that the designer of a system will rarely implement a natural-language dialogue capability. More commonly, a natural-language dialogue package, which has been developed elsewhere, will be selected for incorporation into the system being designed. It is possible, then, that future designers of application systems will be more concerned with selection of such dialogue capabilities than with their design or implementation.

Interactive Graphics

Interactive graphics is not actually a separate dialogue type -- it can, in fact, incorporate any of the dialogue types already mentioned.

Nonetheless, it offers enough unique advantages and problems to warrant a separate discussion. Graphical displays, of course, allow certain classes of information to be communicated to the user at very high information rates, because of the sophisticated visual information processing capability of humans. If they are well designed and sufficiently rapid, graphical dialogues can be extremely "natural" and (from the user's viewpoint) efficient. There is evidence that graphical dialogues are intrinsically motivating, at least for novice users (Solomon & Papert, 1976, discuss this in connection with their use of graphics in computer-assisted instruction for children).

Some application domains are inherently graphical, and are obvious candidates for graphical dialogue. Weinzapfel and Negroponte (undated) suggest that interactive graphics may allow users unskilled in spatial visualization to effectively solve problems which rely heavily on such skills. Graphics can be especially helpful when the problem to be solved has multiple, interacting dimensions. Krolak et al (1971) illustrate a case in which the use of graphics allows a naturally graphical (traveling salesman) problem to be solved using a particular problem decomposition which would be unworkable without interactive graphics. Furthermore, interactive graphics can be very effective in problem areas which are not inherently graphical. Balzer and Shirey (1968) present an interesting example involving the "firing squad" problem. Smith and Crabtree (1975) present a particularly creative application of graphical displays. In their experiment, the user solved a graphical problem involving the fitting of rectangles within rectangles. Their performance was superior to performance on the real, underlying -- and nongraphical -- problem (job-shop scheduling) to which the graphical representation was mathematically isomorphic.

Interactive graphics has a voluminous literature of its own, most of which deals with particular applications or with graphical hardware. In our survey, we attempted to include those documents which dealt fairly specifically with human factors issues and research. Many studies were found which deal with specific issues of graphical input or output

devices and techniques, although research on relatively recent technological developments in graphics was noticeably lacking. These studies are discussed in later sections. Very little research was found which dealt with the higher-level issues of the overall dialogue, however.

Graphics is an area in which research lags application considerably. In this case, the response of the application community has been to proceed cheerfully ahead, making its decisions on whatever pragmatic bases are available. This may be preferable to the alternative -- observed in some areas of software development, for example -- in which a major research gap has been filled largely with unskilled and even haphazard experimental studies.

Among other things, the dearth of quantitative research makes it difficult to assist the designer with cost-performance tradeoffs. Almost any interactive computer application could benefit from a skillful use of graphics, but the performance advantages would, in many cases, fail to offset the significantly higher costs of interactive graphics. Even inherently graphical problems may not justify the use of <u>interactive</u> graphics. In the absence of performance data, tradeoff decisions are very difficult, and are often made for reasons that have little to do with performance.

Modern command-and-control facilities offer some of the greatest dialogue design challenges. Each of these systems typically has multiple users, widely differing in information needs and computer-related experience. Such facilities often involve integration of a wide range of display and input devices, including interactive graphics, usually with restrictions on device choice and use because of security and survivability requirements. There is little information on which to draw in order to design an effectively integrated system of this complexity, although a number of research programs are now attempting to address this problem.

At the level of ordinary interactive graphics at a single workstation, there are several publications which contain useful guidelines or examples. For suggestions on the overall philosophy -- and some details of good graphical dialogue, Foley and Wallace (1974) is an excellent source which emphasizes "natural" communication. Martin (1973) provides less guidance, but contains examples of a wide variety of graphical displays and dialogues which may be suggestive to the designer. Prince (1971) presents a good overview of graphics as applied to computer-aided design.

More detailed guidelines in such areas as display formatting, coding, and input device selection and use are discussed in later sections.

EMBEDDED TRAINING

Embedded training is the use of an interactive system to train its own user. Although it is obviously related to the broader field of computer-assisted instruction (CAI), embedded training occurs only when the subject matter being taught by CAI is the use of the same interactive system for some other primary purpose. Another term commonly used for such systems is "self-tutorial". Embedded training has received attention since the early days of interactive systems, and has found its way into many systems, in greater or lesser degree, but has received limited attention in the form of human factors research or guidelines.

In order to render its boundaries clearer, it may help to distinguish embedded training from some other facilities. Computer-initiated dialogue and ordinary error-messaging and recovery features may help the user learn the system, but they do not constitute embedded training, nor does a simple "help" facility which provides information about system commands on user request. Embedded training involves an explicit tutorial package which teaches the user how to perform the mechanical operation of the system and terminal, or how to utilize the system in the user's

task, or both. It is also appropriate to distinguish embedded training from job performance aids, which may be tutorial in nature, but are intended for use each time the user's task is performed, even after the user is experienced.

Two investigators report explicit attempts to assess the effectiveness of embedded training. Goodwin (1974) describes (with illustrations) a training facility which acquaints new users with a particular terminal workstation. The emphasis in this package is on the mechanical use of the terminal, including display, keyboard, and lightgun. The package is simple, but well done. Morrill (1967) presents a less detailed description of an embedded training package in a management information system. Morrill's tutorial facility teaches not only use of the terminal, but also the dialogue of the system and, to a limited extent, the use of the system in data acquisition.

Each of these studies reports performance data on the use of the system by trainees, but neither provides any sort of comparison with conventional training or any other training approach. In both cases, the data suggest that the training was effective, but are not really definitive without some basis for comparison. Both reports present good ideas and examples for the construction of such packages.

There have been many anecdotal reports that embedded training is effective, and it would be quite surprising if well-designed tutorial packages failed to convey a training advantage in most situations. Learning on the system itself provides a better-than-usual opportunity for high transfer-of-training, and computer systems are ideal for providing rapid feedback, performance tests integrated with the training, etc. In particular, it should be noted that embedded training provides a better training situation than does CAI in general, because of the similarity of the training situation to the ultimate job to be performed by the user.

The surveyed literature contains several examples of much more sophisticated tutorial packages. In general, these are interactive problem-solving or decision-aiding systems with significant tutorial aspects. The tutorial aids are primarily associated with improving the user's decision-making or problem-solving behavior, and so constitute job performance aids or CAI systems, rather than embedded training <u>per se</u>. They do contain useful ideas for embedded training, however. Among these papers are Allan (1968), Baldwin (1977), J. S. Brown et al (1974), Carbonell (1970), Collins et al (1974), and Lumsdaine et al (1970). Caruso (1970) describes a system which nicely integrates tutelage on problem solving in bibliographic search tasks with tutelage on the mechanics of the system itself.

Undoubtedly, the most sophisticated example of embedded training found in the survey is NLS-Scholar (Grignetti et al, 1974). Scholar is a fairly sophisticated, natural-language CAI system which can, in principle, be used to teach in a wide variety of task domains. One of the areas in which it has been applied is in training the user of NLS (oN Line System), a powerful system used for document preparation, communication, and a variety of interactive problem-solving tasks. This may be an important approach in the future. The factors which make natural language difficult to apply in many domains (especially the complexity of the required domain knowledge) are more manageable in the context of teaching an interactive system whose properties are already formally defined.

At present, good (and bad) examples are about all one can point to in the way of a literature on embedded training. However, there is a large literature dealing with CAI, course-authoring languages, etc., which may be relevant. Any guidelines built for the designer in the near future will probably have to rely heavily on general principles for the construction of training materials, with examples of the application of such principles to embedded training.

DIALOGUE SPECIFICATION AND CONTROL TECHNIQUES

The preceding discussion of interactive dialogue has concentrated on the selection of an appropriate dialogue type, the properties which such dialogues should have, and on the possibilities for developing guidelines, on these topics, for use by interactive system designers. The techniques used to specify and to control interactive dialogues are normally considered to be an implementation matter, and are not usually discussed in connection with human factors issues. However, we feel that these topics warrant discussion for three reasons. First, the choice of a dialogue control technique often determines properties of the dialogue. Even with a good high-level dialogue design, undesirable features often creep into the system as a result of this "implementation" decision. The second reason is somewhat corollary to the first. If good dialogue control techniques can improve the behavior of a system -- and perhaps make it easier to implement, as well -- then by making the designer aware of such techniques we may help the designer to produce better systems. The third reason may be somewhat ambitious, given the current state of the art, but it is potentially the most important of the three. As dialogue specification and control techniques become more comprehensive and powerful, it may be possible to "build in" good human factors. At least some aspects of the dialogue might be made to conform to "good" human factors principles without explicit attention by the designer.

Pew and his colleagues (Pew & Rollins, 1975; Pew et al, 1976) have developed a set of dialogue specification techniques whose explicit goal is this kind of "built-in" human engineering. These are not dialogue <u>control</u> techniques -- by which the computer executes the interactive dialogue -- but are methods used by the designer to communicate the desired dialogue to the implementor. The proposed approach represents a very good first step, but covers only relatively simple menu selection and form-filling dialogues. It is not clear that this approach can easily be extended to more complex dialogue, but it can probably be applied to any dialogue types which are basically "frame" oriented (i.e., built around a series of relatively static displays or subdisplays).

It is possible that these frame-oriented specification methods could be extended to dialogue control in such a way that the bulk of the detailed display formatting and input handling would be accomplished automatically and in accord with a reasonable set of *a priori* human factors criteria. This has been done in connection with individual projects, but not, to our knowledge, in any very powerful or generic way. Typically, the capabilities of such dialogue control packages have been strongly influenced by very specific features of the dialogue which they were designed to support.

For question-and-answer dialogues, fairly generic dialogue control software does exist. Carlisle (1972) discusses a language ("HELP") based on constructs similar to IF-statements, and Martin (1973) reviews several simple course-authoring languages, showing their applicability to interactive dialogue control. In the case of these simple Q & A dialogues, display formatting and input processing are trivial, and a specification of the verbal inputs and outputs is virtually a complete specification of the dialogue.

Several generic mechanisms exist for controlling the verbal component of more complex dialogue types. Feyock (1977) discusses the use of state-transition diagrams to describe command languages, and points out that they allow a fairly automatic mechanism for controlling both user-and computer-initiated dialogue, help facilities, and some classes of error messaging and recovery. If the software which controls such facilities already exists, the designer need only specify, via state-transition diagrams, the legal commands. The remainder of the dialogue is determined by the properties of the pre-existing software. It is this kind of approach which offers some potential for "built-in" dialogue characteristics to which the designer need not attend. If, for example, a state-transition diagram is used to control a menu-selection dialogue with a lightpen, such matters as frame formatting, menu placement and spacing, and input processing for the lightpen are not part of the state transition diagram, but of the controlling software.

State-transition diagrams are satisfactory only for the control of syntactically simple command languages. Other mechanisms sometimes used for such languages include IF statements and keyword tables. Kasik (1976) reviews these approaches, and offers a variation on the state-transition diagram, the "triply-linked tree". For syntactically simple, hierarchic command languages, this approach allows automatic processing of the sorts discussed above, and allows level-skipping and some other potentially useful features.

None of these mechanisms has much syntactic flexibility, and they may very well give the designer a feeling of great rigidity and constraint. Two dialogue-control mechanisms exist which are much more powerful and flexible -- production systems and language grammars. These control mechanisms are also much more sophisticated, and may not be useable, in raw form, by more than a small percentage of designers.

Production systems are a familiar tool in artificial intelligence, and have been incorporated as the overall control mechanism for the Rand Intelligent Terminal Agent (RITA; Anderson & Gillogly, 1976). Production systems are very powerful, and are theoretically capable of performing any definable function, but it is not at all clear that they are a natural mechanism for the control of sophisticated interactive dialogues. RITA is impressive in that it can execute complex procedures for the user (e.g., logging onto the ARPANET), but it has not, to our knowledge, been made to interact with the user in other than fairly simple, verbal commands.

Production systems are a relatively specialized and difficult form of programming. Very few interactive system designers currently possess the skills required to use them effectively. It is likely, however, that interactive aids for this programming task may be forthcoming in the next few years. It remains to be seen whether this will be a viable mechanism for the control of "ordinary" interactive systems, or only a tool for research and for the control of highly specialized "knowledge-based" systems.

Although language grammars lack the power of production systems for manipulation of a knowledge base, they are a much more natural mechanism for control of interactive dialogue. The use of a grammar to describe the syntax of a command language or programming language is familiar to most modern designers. By "augmenting" such a grammar with information about the corresponding system actions, the grammar can become a powerful dialogue control mechanism (see Lenorovitz & Ramsey, 1977).

All of these approaches are basically verbal, and require significant extension in order to control graphical dialogues. These extensions include, at the least, such things as display drivers and graphical input processors. For greater transportability, they may include a device-independent graphical language with a translator for the specific display and input devices. This approach has now been developed to the point that its use in operational systems should be commomplace in a few years. Graphical information can be described directly with grammars or similar means (e.g., Carbonell, 1971), but cost-effective operational use of this technique seems to be many years away.

OUTPUT DEVICES AND TECHNIQUES

There is clearly a good deal of interaction between the selection of a dialogue type, design of a command language, etc., and the selection or design of the input/output devices which will be used to implement that dialogue. Decisions concerning these topics are addressed separately here, in order to allow a manageable discussion of the issues. In general, decisions concerning important aspects of the dialogue should precede input/output device decisions. In practice, the opposite often occurs, and we have seen several instances in which the selected terminal devices were incapable of conducting a dialogue acceptable for the task at hand.

As the focus shifts from dialogue to output (display) devices, the available empirical data which can be used to support design guidelines becomes both more voluminous and more directly relevant. A great deal of human factors research has been done on displays and display devices, and this research has concentrated heavily on electronic displays for several years. For some design issues in this area (e.g., display luminance, contrast, refresh rate, etc.,) reasonable guidelines already exist and can simply be incorporated into a more comprehensive design guide. There remain areas in which guidelines are still inadequate or nonexistent.

DISPLAY TYPES

One such area is the selection of the basic type of display device to be used in the system. Although teletypewriters and alphanumeric cathode-ray-tube displays are the most popular choices, numerous other possibilities exist, including plasma displays; LED and liquid crystal displays; tactile displays; audio displays, including automated speech generation; graphical displays, including three-dimensional graphics; laser displays; and even psychophysiological output devices.

Unfortunately, there has been little human factors research on overall performance with most of these display types (see Figure 17).

Display Type	Comments	Principal References
Refreshed CRT	The ordinary, refreshed CRT is currently the "basic" computer display. A good deal of data exists concerning appropriate visual properties of CRT displays. See later discussion of Display Properties. Studies which have compared user performance using CRTs with performance on other display devices do not provide a satisfactory basis for selection decisions (see text).	
Storage Tube CRT	For some graphical applications, direct-view storage tubes may be preferable to refreshed displays. The storage tube allows very high-density, flicker-free displays, but imposes significant constraints on interactive dialogue. Although information exists concerning the basic functional advantages and disadvantages of such displays, no empirical data pertaining to human factors concerns were found.	Steele (1971)
Plasma panel display	Plasma panel displays are inherently "dot", or punctate, displays, and studies of symbol generation method are relevant (see later section on Display Properties). Little empirical information exists on human performance aspects of plasma displays *per se*.	
Teletypewriter	Reasonable guidelines exist with respect to the design of teletypewriter terminals, including both physical and functional properties. See later discussion of Terminals.	Dolotta (1970)
Line printer	Research on typography is voluminous and directly applicable. Research dealing directly with the line printer used in computer output is scanty, but consistent with findings of typographic research (e.g., mixed upper-lower case is best for reading comprehension). Guidelines are not known to exist, but could be constructed with additional survey of typographic research literature. Use of line printers for "pseudographic" displays is common, little discussed in the literature. Pseudographics is an inexpensive way to convey simple graphical information, and should probably be used more widely in batch applications.	Lewis (1972)** Ling (1973) Poulton & Brown (1968)**

Figure 17. Basic Display Device Types (Page 1 of 2)

Display Type	Comments	Principal References
Laser displays	Reasonable human factors guidelines with respect to visual properties have been proposed, but these displays are not widely used.	Gould & Makous (1968)
Tactile displays	Although some tactile displays have been proposed or even developed, little human factors research has been done other than that concerned with prosthetics.	Noll (1972)
Psychophysiological displays	Psychophysiological input is technically feasible now, but psychophysiological displays are still only a topic for research.	
Large-screen displays	There is conflicting evidence with respect to the performance effects of large-group vs individual displays. The main advantages of large-screen displays are a larger display area and the existence of a single display which is clearly the same for all viewers. Unfortunately, higher display content is not achievable due to the resolution limits of existing technology (e.g., light valve displays), and may be unachievable in principle, since the large-screen display usually subtends a <u>smaller</u> visual angle than an individual display located close to the user.	Landis et al (1967)** Smith & Duggar (1965)**
Telephone	See later discussion of Terminals.	
Graphical displays	See later discussion of Graphical Displays.	

Figure 17. Basic Display Device Types (Concluded)

In the case of plasma displays, for example, the only relevant research found in the survey was concerned with punctate versus stroke symbol generation. This topic is discussed in a later section. Nothing was found on light-emitting-diode displays, but they are discussed primarily outside the computer systems literature. Gould and Makous (1968) provide useful recommendations concerning the visual properties of laser displays, although such displays have not lived up to their early promise and are not widely used.

Tactile alphanumeric displays seem to hold little promise, except possibly for blind readers (Shackel & Shipley, 1970), but tactile displays have been proposed for "graphical" information (e.g., Noll, 1972). There has been little human factors research on such displays, at least in the computer-related literature. Psychophysiological displays have been the subject of some research, including even the possibility of stimulation devices surgically implanted over the visual cortex, but no application of even the less esoteric forms appears likely anytime soon.

Computer-generated speech is not yet widely used for real application work, but it is definitely available "off-the-shelf" now. Turn (1974) presents a fairly thorough review of the area, including problems and possibilities. Most human factors research and guidelines applicable to auditory displays are concerned with warning signals and similar matters, but not with automated speech. Witten and Madams (1977) present a study which suggests that even relatively low-quality speech is acceptable for casual users. Smith and Goodwin (1970) make several suggestions for applying automated speech. These suggestions concentrate on the user's ability to pace and control the dialogue and on the use of different voices and tone codes to convey information and indicate required user actions.

The more basic, and unresolved, issue concerns the recognition of tasks in which automated speech is a viable alternative to visual displays. Speech is characterized by low bandwidth, but has the advantage that visual attention is not required. It is thus useful

when the user has some visual task, and is certainly good for warning signals in situations in which operator vigilance is suspect because of competing tasks, long hours, etc. From the literature surveyed, it is not clear that any more extensive guidance can be offered with respect to the use or nonuse of such displays.

Chapanis (1975, 1976) and several colleagues have reported an extensive research program in which two-person teams solved problems while communicating in various media (e.g., voice, video, teletypewriter, handwriting). A consistent finding throughout this program is that problem solving is more effective when voice communication is allowed. However, it is not clear just how this result applies to interactive computer systems. Clearly, most of the advantage is achieved by engaging in a natural-language dialogue which is beyond the capabilities of our current technology. It may also be more important to have automated speech input than output, since visual displays provide a very high rate of information transfer, but input information rates tend to be relatively low with virtually all input media.

Dialogue recommendations pertaining to speech output, such as those offered by Smith and Goodwin, should be incorporated into a design guide, and loose guidelines dealing with vocabulary, phonetic discriminability, etc., could probably be developed based on existing knowledge. Overall, though, the literature does not seem to provide a great deal of support for design decisions involving speech output.

Teletypewriter and cathode-ray-tube (CRT) displays have been subjected to far more extensive empirical study. Reasonable guidelines exist for CRT visual properties, including character generation (see next section). Reasonable guidelines also exist with respect to the physical and functional properties of teletypewriter terminals (e.g., Dolotta, 1970; see later section on Terminals). Even with these well-studied display devices, though, there has been very little empirical study which is concerned directly with the selection of one display device or another. Evans (1976) reports that medical patients prefer

a slow (10 cps), noisy teletype to a CRT terminal for interacting with an automated medical interviewing system. Martin and Chubb (1970) report that users in a logistics management task found CRTs much less boring than a hardcopy device, but no performance differences were reported. Fetter and Carlisle (1971) compared CRT and teletype terminals for use in information retrieval tasks associated with a legal information system. Users took longer, made more errors, and retrieved slightly more information when using the CRT, but the CRT is said to be poor by modern standards.

It seems unlikely that studies at this level will provide any satisfactory resolution to the device selection issue. The utility of various display devices is fairly task-specific, and probably cannot be resolved in such a general way. Guidelines on device selection will probably have to be procedural and analytical in nature, helping the designer to understand the relevant aspects of the selected dialogue method and the user's task(s) and requirements for displayed information. In some cases, such an analysis will clearly eliminate some candidate display devices. In most cases, though, the designer may face a tradeoff decision involving several candidates. At present, the best we can expect of human factors guidelines is that they will inform the designer of the relevant issues.

VISUAL PROPERTIES OF DISPLAYS

Reasonably good guidelines already exist for almost all of the basic visual properties of refreshed CRT displays (see Figure 18) and -- by extension -- of storage-tube and plasma displays, although care is obviously required in making this generalization. The phrase "basic visual properties" is used here to connote those simple display properties, such as luminance and contrast, which are more or less independent of information content and format. Although there is considerable research dealing with these properties, it is important to note that older studies of displays illuminated by reflected light are not particularly relevant here.

Property	Comments	Principal References
Flicker	Perceptible display flicker can cause irritation and visual fatigue, and may impair visual performance. Flicker is a function of regeneration rate, phosphor persistence and chromaticity, and size of display element. Beam scanning sequence (as in interlaced raster scanning) can affect the disturbing effect of flicker, but has only a small effect on perceptibility of flicker. Good guidelines and evaluation methodology exist.	Dill & Gould (1970)** Gould (1968)*
Luminance, Luminance Contrast	Appropriate levels of luminance and liminance contrast may be affected by room illumination and display chrominance. Good guidelines exist.	Gould (1968)*
Chromaticity	Chromaticity of single-color displays is probably not a major issue. The yellow-green portion of the spectrum corresponds to the greatest sensitivity of the eye, but adequate luminance is the issue there. Possible effects of chromaticity on eyestrain is an open issue. See discussion under "color coding" for information pertaining to multicolor displays.	Gould (1968)* Mezrich et al (1977)**
Resolution	Some guidelines exist with respect to display resolution, but they vary with visual task, type of display element, display element size, etc. For alphanumeric characters, raster displays should provide at least 8 and perhaps 10 scan lines per character height, although the required number varies inversely with character height over a significant range. Dot matrix displays should use at least a 5 x 7 matrix, and preferably 7 x 9. Construction of more extensive guidelines considering visual task, etc., is possible, but would turn up numerous gaps. Good techniques exist for evaluating legibility of specific display.	Buckler (1977) Gardner & Soliday (1974)** Hemingway & Erickson (1969)** Smith & Goodwin (1973)** Vartabedian (1970a)**

Figure 18. Basic Visual Properties of Displays

Flicker

Perceptible display flicker can result in irritation and fatigue, as well as perceptual problems (Oestberg, 1975). In displays which are refreshed at a constant rate regardless of content (e.g., standard raster displays), flicker is basically a hardware design issue. In refreshed vector graphical devices, display element type and density affect display writing time, which can affect regeneration rate and thus cause flicker. Flicker may be a hardware or software issue here.

Fairly good quantitative information exists relating flicker to regeneration rate, angular size of display element, luminance, and phosphor persistance and spectral composition (Gould, 1968). Dill and Gould (1970) supplement this information and investigate the effect of beam-scanning sequence, which turned out to be smaller than expected. Good guidelines exist with respect to the regeneration rate required to avoid flicker with various phosphors, and good empirical methods of assessing the acceptability of a particular combination of values are available.

Luminance and Luminance Contrast

Gould (1968) provides a good review of the factors affecting luminance and luminance contrast. Luminance contrast is also discussed fairly well by Bryden (1969). Gould's recommendations concerning these factors appear sound. Oestberg (1975) and Stocker (1964) point out that office and background illumination affect desirable display luminance values. Oestberg, as well as Hultgren and Knave (1974), feel that "contrast glare" due to use of terminals in bright offices is an important factor in operator eyestrain and fatigue. Hultgren and Knave present simple guidelines for dealing with this problem, and Stocker suggests a method for arriving at appropriate display and background illumination levels when the user must work with both displays and hardcopy. Mezrich et al (1977) point out that

luminance interacts with chrominance (hue) because of the nonlinear spectral sensitivity of the visual system. They present a quantitative model which could probably be useful in correcting luminance levels for changes in phosphor chrominance. Empirical evaluation is preferable, however, when a new display device is being developed.

Chromaticity

Chromaticity is probably not an important issue in monocolor displays, as long as luminance is adequate (Gould, 1968). Mezrich et al (1977) present a model of the relationship between these factors. Although the greatest sensitivity of the eye is in the yellow-green region, most alphanumeric CRT displays produced in the U.S. are white. In Europe, there is a tendency to move toward yellow on the theory that it reduces eyestrain, but we are unaware of empirical evidence specifically supporting this decision.

Resolution

Most studies of display resolution have dealt with alphanumeric or geometric symbols and have used either standard visual acuity measurement techniques or visual search tasks. The use of measurement techniques more nearly similar to those which will be performed by the user may be more appropriate. For example, Smith and Goodwin (1973) propose a check-reading task in which the reader searches for random letter substitutions in connected text. This technique is applicable to the measurement of legibility as a function of any display parameters, but is mentioned here because such techniques are most often applied to studies of resolution.

Reasonable guidelines exist for the most common questions of display resolution (e.g., number of scan lines or dots per character). Even here, reasonable discretion is required in generalizing to tasks other than those studied, and in correcting for effect of variations in other display variables. More detailed guidelines providing for

such factors could be developed, based on existing information. They would necessarily have gaps, and should in any case be supplemented with empirical testing of the legibility of the particular device.

Properties of Alphanumeric Characters

A very large number of studies have been done pertaining to the properties of alphanumeric displays. Most of the studies do not deal with electronic displays, and were outside the scope of this survey even though they may be relevant to the design of such displays. It is worth noting, however, that the visual properties of luminous displays are somewhat different from those of hardcopy displays illuminated only by reflected light.

Important properties of alphanumeric characters include generation method (raster, dot, stroke), font (character shape and number of elements), size, case (upper case only, or mixed upper and lower), spacing, and aspect ratio (ratio of height to width). These variables are discussed in Figure 19.

The best overview of alphanumeric display properties discovered in the survey is Buckler (1977). This paper contains a well integrated set of general (not task-specific) guidelines for the variables mentioned above and others.

DISPLAY CODING TECHNIQUES

A large number of empirical studies have been done relating to alternative display coding techniques (color coding, blink coding, etc.). Most of these studies involve a comparison of two or more different coding techniques in the context of a particular task. This information may be helpful in selecting a coding technique. Few of the studies seriously attempt to help the designer decide exactly how to apply the chosen coding technique, however.

Property	Comments	Principal References
Generation method	For visual search, and perhaps reading, dot-matrix characters appear to be slightly preferable to stroke-generated characters. Resolution and font are related issues. In particular, if individual dots are easily perceptible, small area flicker can occur, and there is evidence that characters composed of perceptible dots may be encoded in human memory in a special way. This might affect performance on matching tasks, for example.	Buckler (1977) Maddox (1977)* Peters & Barbato (1976)** Vartabedian (1971)**
Font	Simple character fonts are best, with no serifs, variable stroke width, slanting, etc. Dot-matrix characters using a 7 x 9 dot matrix yield better performance than with 5 x 7 format, but 5 x 7 appears satisfactory for most tasks. The standard Leroy font (and similar MIL-M-18012) appear satisfactory for computer displays, as well as hardcopy.	Buckler (1977) Vartabedian (1970a)**
Size	Desirable character size (measured in angular subtense) varies with visual task (e.g., larger characters are required for single character search and recognition than when whole words are involved). Required character size varies inversely (within a limited range) with display resolution, at least in raster displays. Reasonable guidelines exist.	Buckler (1977) Giddings (1972)** Hemingway & Erickson (1969)** Whitham (1965)
Case	Evidence seems to indicate that the use of all upper-case characters is preferable for visual search tasks involving whole words, but that ordinary mixed upper and lower case is better for reading tasks.	Poulton & Brown (1968)** Vartabedian (1971)**
Spacing	Data from non-electronic displays suggest that intercharacter spacing should be 26-63% of character width. Virtually all electronic displays use uniform spacing, rather than the proportional spacing used in typesetting, but no data were found indicating the performance effects of this practice.	Buckler (1977)
Aspect ratio	Symbol height-to-width ratio does not appear to be a significant parameter. A range of 1:1 to 2:1 seems acceptable. On non-electronic displays, a ratio of 4:3 has been found most legible.	Buckler (1977)
Cursor	On an alphanumeric display, the cursor marks the character position at which changes, insertions, etc., will occur. Several cursor forms seem to be acceptable based on visual search and cursor placement times, and user preference ratings. Acceptable types include a box, diamond, cross, or underline blinking at 2-5 Hz. The cursor should not blank out or modify the character it marks. Nonblinking cursors yield inferior search performance. Best overall performance, of those tested, was a box blinking at 3-4 Hz.	Vartabedian (1970)**

Figure 19. Display Properties of Alphanumeric Characters

Many of the studies are relatively old, and address coding in other contexts than computer displays. Their generalization requires some care, since existing computer displays involve images which are composed of (sometimes perceptible) dots or lines rather than continuous areas of constant hue, brightness, etc.

Nonetheless, there is a wealth of relevant information on this topic which can reasonably be integrated into a design guide. In the case of color and blink coding, the literature included in the survey is probably adequate for this purpose. In other cases, reference to the older, non-computer-specific literature would be required. Figure 20 outlines some major conclusions with respect to the four most common coding techniques.

Alphanumeric Coding

The survey turned up very little information on alphanumeric coding which is either recent or specific to computer displays. Older studies referenced by Christ (1975) and by Grether and Baker (1972) indicate that alphanumeric coding is the most accurate technique for identification tasks, and may be tolerable for search tasks. It also has the advantage of allowing a practically unlimited number of coding categories. The older studies contain performance data for search time, etc. There are several specific issues to be addressed if alphanumeric coding is to be used satisfactorily, as indicated in the figure.

Shape Coding

Shape coding, as by the use of geometric symbols, is widely used, particularly in graphical displays. Because the symbols are often designed for a particular application, performance data are usually not generalizable. Research does indicate that shape coding is useable for visual search and identification tasks, but that other coding methods yield better performance. Grether and Baker (1972)

Coding Method	Comments	Principal References
Alphanumeric coding	Alphanumeric coding is best for absolute identification, but may involve problems of size (coding usually is accomplished by adding additional symbols, which take up space on the display), confusability of similar symbols, learning of associated meanings, and sometimes superimposition of coded symbols.	Christ (1975)* Grether & Baker (1972)
Shape coding	Shape coding is useable for both search and identification tasks. The use of meaningful shapes or geometric symbols is widespread, but has received little empirical study. A key issue is symbol discriminability.	Christ (1975)* Grether & Baker (1972)
Color coding	Relevant color coding -- redundant or nonredundant -- generally yields better performance than other coding methods in both search and identification tasks, except that alphanumeric coding yields better identification accuracy. There are minor exceptions, however, and performance advantages of color are much greater in some situations than in others. There are a number of other factors which might influence the selection of color as a coding mechanism. Users tend to prefer color even when there is no performance advantage, and possibly when the overall effect on performance is negative.	Christ (1975)* Teichner et al (1977)**
Blink coding	Blink coding is effective as an inclusion or exclusion code for target detection tasks, but can adversely affect reading performance if user cannot match scan rate to blink rate. Blink coding helps most with high density displays. Although users can discriminate up to four blink rates, blink coding should probably be restricted to a binary code (1 class flashing, 1 static). In this case, optimal flash rate is probably 3-4 Hz. No research is known on optimal on-off cycle.	Smith & Goodwin (1971, 1972)**
Others	See text.	

Figure 20. Major Display Coding Techniques.

recommend that not more than 15 symbols (preferably not more than 5) be used.

It should be noted, however, that most studies of shape coding have involved abstract tasks, without an underlying, meaningful application, and symbol sets constructed primarily for visual discriminability. That is, the users had no prior association of the symbols with task-relevant entities.

In those situations in which the association of symbols to targets is not intended to be meaningful, the data derived from earlier experiments are probably applicable. Furthermore, there exist standard "alphabets" of symbols, already optimized for discriminability, which can be recommended to the designer. Some care is required in applying these symbol sets to computer displays as the symbol generation method (vector, dot, raster) may influence the discriminability of the symbols.

There are many situations, though, in which the user's prior association of symbols with meanings is a primary advantage of shape coding. In some cases, standard symbol sets are adapted for use on the system. For example, there are several relatively standard sets of military unit symbols used in tactical displays at various echelons. In other cases, no pre-existing symbol set is used, but symbols are designed whose pictorial properties suggest their meaning. In these situations, the data from classical research studies are of limited assistance. Particularly in identification tasks, it is likely that many more code levels are possible here than with meaningless symbols. A design guide should perhaps make suggestions which will assist the designer in maximizing discriminability of such symbols without losing symbol recognizability.

Jacob et al (1976) have shown that the use of meaningful symbols may be advantageous even in situations in which symbol meaning does not correspond to the properties of the target. By using symbolic

faces with various facial expressions, instead of geometric shapes or several other codes, they have demonstrated enhanced performance on both sorting and paired-associates tasks.

Color Coding

Like many aspects of interactive system design, the effects of color coding vary with a number of other task and display factors. In the case of color coding, however, there have been enough studies that we can begin to see the pattern of these interactions. Christ's (1975) excellent review of these studies identifies several task and display coding factors which interact with color.

In general, relevant color coding -- redundant or nonredundant -- yields better performance than other static achromatic coding techniques in visual search tasks. In this context, "relevant" implies not only logical relevance, but also prior knowledge, by the user, of the color of the target item. When the task is to identify the target, identification by alphanumeric code is more accurate, but color is preferable to other static achromatic coding dimensions. Performance advantages are quite small in many of these situations, however, and may be negated by interference with other coding dimensions, as well as cost and other factors discussed below. When targets are color coded in a way that is irrelevant to the current task -- as might easily happen when different coding methods are used to signify different target properties -- both search and identification performance are degraded. Christ's review contains quantitative data on performance effects which might be useful to the designer in making tradeoff decisions among coding dimensions, especially when more than one coding dimension is to be used.

Teichner et al (1977) present a very good analysis of color coding, and suggest that it is most effective when: "(1) designating a specific target in a crowded display, (2) demarcating an area of a display, (3) providing warning signals or commands which have a limited number of possible alternatives, and (4) classifying or grouping data where the number of classifications are small."

Grether and Baker (1972) provide guidance with respect to the number of hues (maximum 10, recommended 3), and specific hues, which might be appropriate for color coding. Haeusing's (1976) data deal particularly with electronic raster-scan displays, and are probably more reliable in this context than the older studies used by Grether and Baker, but Haeusing deals only with absolute color judgment tasks. Situations in which colored symbols may overlap (yielding different hues) are treated by Smith (1963).

However, there are a number of factors other than simple psychophysical data which might influence these choices. Some color displays have a restricted color spectrum. Beam penetration color CRT displays, for example, are presently restricted to hues ranging from red to a somewhat yellowish green. For situations in which spatial position is important, these displays may also have rather critical calibration problems. Chromatic aberration can even induce undesired three-dimensional effects which influence the user's perception of spatial position. Existing vector graphic color displays are generally rather limited in display capacity, while raster displays may have visual resolution problems. The decision to use color, then, can have a variety of side effects.

There is evidence which suggests that users have a rather strong preference for color displays, even when the use of color carries no performance benefits. This may lead to an overuse of color displays in situations in which the use of color coding to aid in search or identification is unnecessary. There are no known data which would help in weighing possible long-term motivational benefits of color displays against cost and other disadvantages in such areas. At the very least, though, such possible motivational benefits are somewhat speculative.

Blink Coding

Smith and Goodwin (1971) investigated the use of blink coding in a target detection task. They found a 50% improvement in search

times when all items of the target class were blinked at a 3-Hz rate. When all nontargets were blinked, performance improved almost as well. In a later experiment involving a check-reading task (Smith & Goodwin, 1972), reading performance was found to be slightly degraded. The authors suggest that avoidance of such degradation probably depends on the user's ability to adapt the visual scan rate to the blink rate of the system. This depends on the task, and might suggest that the blink rate should be determined in part by an analysis of the "natural" scan rate of the task.

Although there is evidence that users can discriminate up to four different blink rates (Cohen & Dinnerstein, 1958), we believe that blink coding should normally be restricted to two categories (one non-blinking) with the possible addition of a third for some special purpose (e.g., a cursor). Multiple blink rates on a display can be irritating, and perhaps fatiguing, if not directly disruptive of performance.

For simple visual search tasks with only one item or item class blinking, a blink rate of 3-4 Hz is probably best (see Smith & Goodwin, 1971; Vartabedian, 1970; for indirect evidence). No research was found dealing with on-time versus off-time in the blink cycle, but on-time should probably be 50-60% of the cycle.

<u>Other Coding Techniques</u>

Several other coding techniques exist which are less frequently used than those discussed above. These include:

Size
Depth
Line type (solid, dashed, etc.)
Brightness, line width
Motion
Focus or distortion

Guidelines for the use of these coding techniques can suggest the maximum number of coding levels and point out interactions (e.g.,

brightness and line width should not be used together on CRT displays). In general, though, the only empirical data available on these techniques are simple psychophysical data on recognition and discrimination. There is little research bearing on the application of these coding methods to interactive systems.

INFORMATIONAL PROPERTIES OF DISPLAYS

Display Density

The amount of information, or density of information, which a display should have involves both perceptual and cognitive issues. It has been shown repeatedly that increasing the number of displayed items (relevant or irrelevant) on a display increases time and errors in such tasks as visual search, counting, noting of display changes, and extracting information from changed items (Baker et al, 1960; Bowen et al, 1963; Callan et al 1977; Coffey, 1961; Earl & Goff, 1965; Green 1953; Poulton, 1968; Ringel, 1969; Ringel & Hammer 1964; Ringel & Vicino, 1964; Schutz, 1961a). These studies provide no "optimal number of display elements", since that is highly task-specific, but they do provide data relating time and error performance to the number of displayed items, especially for visual search tasks. They also suggest system design actions which can help alleviate problems of high display density.

Obviously, differential coding of relevant and irrelevant classes of display items can help (see especially Ringel, 1969). More to the point, though, is the mimimization of the number of items which are coded with the relevant class code (see, for example, Poulton, 1968). Reduction in number, or total elimination, of members of irrelevant classes can also help significantly. Stewart et al (1974) suggest that the user should have the capability of manually, and reversibly, eliminating such items from the display.

It is also clear that people have subjective preferences with respect to the amount of information present in a display, with subjective ratings typically declining as the amount of information deviates from the preferred amount, in either direction (e.g., Vitz, 1966).

Aside from purely perceptual problems, a high density of displayed information can also cause a "cognitive overload", with effects not only on speed, but also on the quality of the user's information-integration and decision-making performance. Dorris et al (1977) varied both the number of "data sources" in a display and their informational value (in the sense of correlating with a criterion measure) and assessed the ability of people to integrate the information from the various sources in order to predict the criterion value. Performance improved as the number of data sources increased from two to five, and then dropped off when eight data sources were displayed. Perhaps more important, performance was uniformly poor when the data sources were uncorrelated with one another. Like many other studies, this experiment illustrated the rather sharp limits of human ability to integrate information from a variety of unrelated sources. Unfortunately, guidelines on this topic are not easy to generate, and we do not immediately see how to do so. In complex cognitive tasks, a proper task analysis and consideration of appropriate problem-solving aids -- both of which were discussed earlier -- may help. Whether or not guidelines at the level of display formatting will help is an unresolved issue.

Display Formatting in General

Standard human factors textbooks and guidelines contain a great deal of material dealing with displays. Only a portion of this material is applicable to typical computer displays, however. For example, Grether and Baker's (1972) very good overview of human factors issues in displays deals largely with analog displays such as those in aircraft, but has little information on either textual or graphical displays. Although there is a good deal of relevant

research, no extensive guidelines appear to have been developed for displays of this sort. Engel and Granda's (1975) presentation is excellent, but has many gaps, and Danchak's (1976) discussion is nicely popularized, but sketchy and, in places, unsound.

Stewart (1976) provides a brief overview of several basic principles which the display designer should keep in mind. These principles include logical sequencing, spaciousness, relevance, consistency, grouping, and simplicity. An example is provided which illustrates the nature and benefits of these principles rather well. It is not entirely clear that this is sufficient information to allow the designer to successfully apply the principles, however. Certainly some such discussion, which provides a conceptual framework for more specific guidelines (e.g., Engel & Granda) is appropriate in a design guide. It may be that textbooks on graphic arts and presentation media are the best existing source for such a high-level introduction, since psychological research has not yet succeeded in producing a unifying conceptual framework in this area, as will be seen.

Several attempts have been made to determine what aspects of displays are of primary psychological importance. A successful effort here could provide a very useful conceptual framework around which to orient both display formatting guidelines and display research. Thus far, the efforts have not been totally satisfactory, though. The best of these efforts is probably the factor-analytic study of Siegel and Fischl (1971) and the resulting "Analytic Profile System" (Siegel et al, 1969). This effort identified seven subjectively relevant dimensions of the displays used in the study. The dimensions were: stimulus numerosity, primary coding, contextual discrimination, structure scanning, critical relationships, cue integration, and cognitive processing. There are a number of methodological criticisms of this study which make these factors somewhat suspect, and they are, as stated, based on subjective measures rather than performance. They have good face validity, though, and are well explained.

Landis et al (1967) attempted to use a multiple linear regression to relate display and operator properties to operator "decision quality". Their attempt has the advantage that it is based on performance, rather than subjective ratings. They found a number of specific relationships between display properties and performance (e.g., small displays were better than large, non-redundant color coding helped, etc.). Their findings are probably quite task-specific, however.

Given the present stage of development of such studies, a generic framework does not exist which can encompass a significant variety of display or display element types in a way useful to the designer. Thus, the guidelines must either provide only a general overview or must provide a large number of fairly situation-specific suggestions. Indeed, it may be impossible to avoid the latter even with a good framework, since many guidelines (e.g., display of dates and times) may be matters of population stereotype rather than good generic principles.

Two other aspects of display formatting are largely independent of detailed display type, and should be covered prior to a discussion of individual display types. The first is display partitioning, which involves the sharing of a single display surface by several display areas, usually segregated by lines. Although this is a common practice which is clearly advantageous in many cases, no research is known pertaining to methods of partitioning, number of areas, etc. Miller and Thomas (1976) make some reasonable suggestions for use of various partitions. Martin (1973) shows numerous examples of partitioned displays.

In the event that the user must have access to more information than can reasonably be presented at one time, the display may provide access to a selectable portion of the information (as by windowing or scrolling), or the display may be time-shared so that only a portion of the information is displayed at once, with the user able to switch the display back and forth in order to see it all.

Although these techniques are in common use, little research has been done on them.

In windowing, the display is used as a "window" through which the user can view a moveable portion of the display space. Windowing can be used with graphical or tabular displays, as well as alphanumeric displays which convey spatial information (e.g., some air-traffic-control displays). Typically, the user has left, right, up, and down controls with which to move the window or the display. Granda (1978) has demonstrated the feasibility of windowing in a visual search task, but was unable to resolve the "inside/outside" problem of whether the direction of the user's controls should correspond to the motion of the window or of the underlying display. Our very informal study of this problem suggests that pilots and non-pilots have opposite stereotypical responses in this situation, but we have not performed a controlled experiment.

Scrolling is similar to windowing, but involves only text, and only vertical motion of the window. No research was found dealing with scrolling. Giddings (1972) presents an interesting discussion of the time-sharing of displays, and makes suggestions with respect to determining what information can be time-shared, preventing the loss of positionally coded information, and providing appropriate means for operator control of time-shared displays.

It is fairly typical for sophisticated graphical applications to involve partitioning, windowing, and time-sharing, and these features can interact. A discussion of these topics in a design guide should, therefore, be integrated. At present, though, the content of such a discussion will depend more on art than on science.

Graphical versus Nongraphical Displays

In many instances, the designer has a choice between presenting information in graphical form and presenting it in tabular

or other alphanumeric form. Although the merits of these different approaches are by no means settled, there are a few studies which bear on this decision. Important factors are the relative importance of speed vs accuracy, whether or not the user is required to recall information from the display after it has been removed, and, of course, the nature of the information to be conveyed.

There is evidence that information which is spatially encoded on the display can be recalled with greater speed and accuracy than information which is alphabetically encoded (Howell & Tate, 1966; Newman & Davis, 1963), although a study by Nawrocki (1972) failed to substantiate this finding. There are many task variables not explicitly considered in these studies, and not enough research has been done to allow a firm conclusion. There is, of course, a large literature dealing with verbal versus nonverbal encoding of information *in human memory*, and that literature was not included in this survey, nor is that literature well-integrated (or even consistent).

Wright and Reid (1973) compared flowcharts, decision tables, prose, and "short sentence" descriptions of a simple algorithm in an experiment in which participants hand-executed the algorithm and memorized it. The flowchart -- which *might* be correctly viewed as a graphical display -- was superior for hand execution, but was not conducive to memorization. The "short sentence" form, a structured verbal presentation similar to a program design language, was best for memorization. It is not entirely clear, though, that a flowchart should be regarded as a graphical in the same sense that cartographical, pictorial, and Cartesian-graph displays are graphical.

A little more information is available for those situations in which the user extracts information from the display while it is still present. When the information to be displayed is a mathematical function, it would appear that graphs allow more rapid extraction of information about individual values of the function, but tables allow equal accuracy (Cropper, 1968) or greater accuracy (Carter,

1947, 1948; Hoisman et al, 1963). The accuracy question is greatly affected by the need for interpolation. This varies with the task, size of the table, and -- for functions involving three or more variables -- the number of lines on the graph. If also seems clear that graphical presentations are preferable if the user's task is to understand the basic form of the functional relationship without regard to quantitative accuracy.

Booher (1975) compared the effects of pictorial and verbal information when the user's task was to execute a procedure involving a series of discrete actions in the use of electronic equipment. Speed of execution of the procedure was greater when the procedure was described pictorially, but a verbal description of the procedure resulted in greater accuracy. At a superficial level, this result appears similar to those discussed above. It is likely, though, that a detailed task analysis would show that the pictorial information aided in some subtasks (e.g., locating a switch or dial), while the verbal information was more helpful in others (e.g., achieving the correct dial setting).

Vicino and Ringel (1966) studied the effects of graphical and alphanumeric presentations of tactical information used by decision makers to recognize that enemy forces were massing for attack. There was no difference in decision quality between the two presentation forms. It should be noted, however, that the study used personnel unfamiliar with this task and with the symbols involved. This result might very well be different with experienced users.

Overall, the research literature dealing with the graphical-nongraphical display question is disappointing. Not only is more research needed before extensive guidance can be confidently provided, but that research must be much better integrated than it has been in the past. Otherwise, variations in task and type of information to be displayed will render the guidelines inapplicable to most design decisions.

Formatting of Tabular Displays

The information available with respect to the formatting of tabular displays does not seem to have changed much since Stewart et al's (1974) survey. Most of the studies in this area did not deal explicitly with computer displays, and were, therefore, excluded from this survey. We are aware of them from secondary sources, however. The available empirical information is sketchy, and it may be that the best source of guideline material is still to be found in the existing guidelines for graphic design used by artists, publishers, etc.

Several investigators have studied the differential effects of horizontal and vertical arrangements of lists. When no recall was required, Coffee (1961) and Earl and Goff (1965) found no differential effects. Mayzner (1966) and Mayzner and Gabriel (1964) found that spatial organization of alphanumeric data is important when the material is to be recalled. Cropper (1968) demonstrated, among other things, that breaking tabular displays into blocks (e.g., 3 x 3 blocks) improved visual search time. Wright (1968) investigated several different tabular formats (e.g., matrix, "schematic" table, table of lists) and found significant performance differences.

Haney (1969) compared a well-formatted tabular display -- containing test instructions for electronic equipment -- with the previously used verbal description of the procedure. The tabular display resulted in fewer errors. This report provides several ideas for the tabular presentation of procedural information.

Stewart's (1976) formatting principles, discussed above, are presented primarily in the context of tabular displays, and are highly applicable here. Engel and Granda (1975) also present a number of low-level guidelines for tabular displays, and especially list displays. However, no well integrated set of guidelines was found.

Formatting of Graphical Displays

A very similar situation exists in the area of formatting graphs and charts. Almost all of the research literature on this topic is concerned with non-computer displays, and was, therefore, not within the basic scope of the survey. This research literature contains many studies comparing two or more kinds of graphical displays for the presentation of a particular class of information. Many of the studies were done more than 20 years ago, and they do not form a particularly well integrated body of knowledge, but they are relevant to the development of design guidelines in this area. Studies of which we are aware include Bowen and Gradijan (1963), Cropper (1968), Croxton & Stein (1932), Price et al (1974), Schmid (1954), Schutz (1961a, 1961b), Schwartz and Taylor (1968), Strickland (1938), Vernon (1946, 1950, 1952a, 1952b, 1953), Wainer (1974), and Washburne (1927).

It is clear that many specific guidelines can be derived from these studies. For example, multiple lines on a single graph are better than multiple graphs if the task requires comparison of the trend lines, but not otherwise (Schutz, 1961b). It is also clear, though, that this approach will leave many gaps in the resulting guidelines. *It would appear that the best approach* is to begin with existing graphic-arts design material, which is clearly pragmatic in origin, and rework it in accordance with available empirical data. The dangers of "folklore" are well established, and care will be required in exercising this approach. Some existing guidelines in the graphic arts area which might provide a good starting point are Lutz (1949), Schmid (1954), and possibly Huff (1954), Bertin (1967), and Dickinson (1973).

Modern applications of interactive graphics tend to involve much more sophisticated displays than simple charts and graphs, however. A few empirical studies have been done on the graphical display of particular kinds of information (e.g., geographic,

hierarchic, temporal -- see next section on special display types), but no well-integrated picture emerges from the surveyed literature. This is an area in which applications are being developed with relatively little human factors research input, although human factors personnel seem to be able to assist significantly when they are involved in such projects, on the basis of their general knowledge of human factors principles and of human vision and information processing. Unfortunately, it is not yet clear how to capture this knowledge in a way that is really useable to the designer, other than by providing guidelines for specific display types, coding, etc.

Special Display Types

Some useful information is available concerning effective designs for a few special display types (e.g., geographic displays). Much of this information is to be found in reports of the design of particular systems, which were not included in the survey. A number of research projects have also investigated specialized display types, and those reports were generally within the scope of the survey. It seems reasonable to suppose that guidelines dealing with particular display types would be effective in a design guide, especially if the design guide is aimed at a particular class of system. Effective development of such guidelines would require additional attention to the literature on specific systems, however. It would also require considerable reliance on general human factors knowledge, since available empirical data cover only a small proportion of the design issues. Much of the guidelines development effort would involve evaluating ideas used in existing systems or advocated in the literature.

Geographical displays, especially for tactical systems, have received considerable attention (see especially Bowen et al, 1975; Connelly, 1977; Hammell et al, 1975; Irving et al, 1977; Smith et al, 1972; Vicino & Ringel, 1966). Displays for hierarchic search

dialogues are discussed by Thompson (1969, 1971) and Uber et al (1968). TOTE displays were investigated by McKendry et al (1969). Displays of mathematical trends were investigated by Schutz (1961a, b), but see the previous discussion of graphical display formatting for other papers dealing with graphs and charts. Temporal displays which present information on a series of discrete events are discussed by Chesler and Turn (1967), and are discussed in the specific context of process control by Goodstein (1969) and Griffin (1973). The use of time compression in displays for monitoring tasks is shown to be effective by Howell et al (1966) and Scanlan (1975), who present useful ideas for employment of this technique. Three-dimensional displays are discussed by Guttmann and Anderson (1962) and by Vlahos (1965), who provides a reasonably good discussion of the visual cues and "anti-cues" involved in such displays.

INPUT DEVICES AND TECHNIQUES

INPUT DEVICE TYPES

There exist a large number of input device types which might be considered by the designer of an interactive system (see Figure 21). Although there is a very large body of research reports in this area, the vast majority of the research has been concerned with keyboards, and most of the other input devices have received relatively scant attention. The literature on keyboards is quite rich, and provides a reasonable basis for guidelines on almost all physical aspects of keyboards, and many logical and procedural aspects as well. The research dealing with other input devices, at least in the context of computer systems, consists mostly of studies comparing the speed and accuracy of two or more specific input devices in the context of a particular task. While such studies may help in device selection decisions, they provide little data to support the design of the input device itself.

It is useful to consider alternative input devices in terms of five main types of input tasks:

(1) Text input
(2) Input of numerical quantities
(3) Selection of command, operand, etc. from display
(4) Discrete positional ("graphical") input
(5) Continuous positional ("graphical") input

This is not necessarily a complete list; for example, it is not entirely clear where the selection of binary values or the input of alphanumeric codes belongs. It appears, though, to be an appropriate list for briefly discussing the properties of the major interactive input devices.

The virtual input devices (pick, button, locator, valuator) advocated by Foley and Wallace (1974) and Wallace (1976) provide a somewhat similar view of input task types, and suggest that the variety of input tasks could usefully be reflected in the modular structure of interactive dialogue control software.

Input Device	Comments	Principal References
Keyboard	The vast majority of past research on input devices has dealt with keyboards. Reasonable and fairly detailed guidelines exist with respect to the physical properties of keys and keyboards and -- to a lesser extent -- their layout, logical properties, operating procedures, etc. Guidelines for alphabetic keyboards are particularly good, and those for numeric keypads are reasonable. Function keyboards are rather system-dependent; guidelines can specify their physical properties, but can only suggest methods and basic principles for function selection and layout. It is not clear that chorded keyboards are viable except in highly specialized situations.	Alden et al (1972) Seibel (1972)
Lightpen, lightgun	*Lightpens can be used effectively for cursor placement and text selection,* command construction, and for interactive graphical dialogues in general, including drawing. However, there is evidence that greater accuracy may be possible with a mouse in discrete tasks, and with a trackball in drawing tasks. Mode mixing, as by alternating use of lightpen and keyboard, can significantly disrupt performance, since the lightpen must be picked up and replaced with each interval of use. Continuous use of a lightpen, at least on commercially available CRT terminals with vertical display surfaces, can be quite fatiguing. There has been no known research on desirable physical and logical properties for lightpens.	English et al (1967)** Goodwin (1975)** Irving et al (1976)** Ramsey (unpublished)**
Joystick	There are many studies of the use of joysticks for continuous tracking tasks, but few studies of its use for discrete or continuous operand selection or graphical input tasks. Those studies which have been performed have found the mouse, lightpen, and trackball preferable in terms of speed, accuracy, or both. Joysticks are sometimes used for windowing and zooming control in graphical displays. No research on this topic was found, although the results of tracking studies may be applicable here. Otherwise, no clear recommendations for joystick properties emerged from the survey, even with respect to basic issues like position vs. rate vs. acceleration control. These issues may be fairly task-specific.	Card et al (1977)** English et al (1967)** Irving et al (1976)**
Trackball	*The trackball appears to be effective for both discrete and continuous operand* selection and graphical input tasks, and may yield the best performance when graphical inputs must be alternated with keyboard input. No empirical data on physical properties were found, but some such data are thought to exist in the tracking literature.	Irving et al (1976)**

Figure 21. Input Device Types (Page 1 of 3)

Input Device	Comments	Principal References
Mouse	Although the mouse is not in widespread use, there is evidence that it is an effective device for text selection. No data are known concerning its physical properties, or its use in other tasks.	Card et al (1977)** Engelbart (1973) English et al (1967)**
Graphical input tablet	Graphical input tablets are capable of fairly high pointing accuracy (within 0.08 cm, according to one study). They are commonly used for freehand drawing, but may be inferior for discrete position input tasks. They may also involve a performance decrement due to low stimulus-response compatibility when the drawing surface is separate from the display surface.	English et al(1967)** Myer (1968)**
Touch panel	No empirical performance data were found dealing with the touch panel. While its inherent resolution limits may preclude serious use for fine discrete position and continuous position input, it feels "natural" and may become a common device for more coarse positioning and selection from lists.	Hlady (1969) Johnson (1977)
Knee control	A knee control has been used in one research study for discrete position input. It is not known to be in use otherwise, and seems unlikely to see serious use.	English et al (1967)**
Thumbwheels, switches, potentiometers	These have been studied primarily outside the computer systems domain, and are discussed in standard human factors reference sources. They are not often used as major user input devices for interactive computer systems.	
Tactile input devices	Although some tactile input devices have been proposed, little human factors research has been done other than that concerned with prosthetics.	Noll (1972)
Psychophysiological input devices	Electromyographic signals have provided superior performance in some control tasks to joysticks and other manual control devices. Use of heart rate, keyboard response latency, EEG input, etc., is technologically feasible, although really sophisticated input is not yet achievable via these methods. There are ethical and legal problems here, as well as technological difficulties. Significant human factors data were not found with respect to computer-related use of these techniques.	Slack (1971) Wargo et al (1967)**

Figure 21. Input Device Types (Page 2 of 3)

Input Devices	Comments	Principal References
Automated speech recognition	The current state of this technology limits its use to relatively simple input tasks. Even there, there are problems with different speakers, noise, etc. Although speech input seems like a very desirable and natural input node, and is clearly preferred over other communication modes for interpersonal communication, it is not clear whether it will prove to be widely applicable for human-computer interaction tasks. Very little information was found which would assist the designer in recognizing tasks for which speech input is appropriate, or in selecting an appropriate speech input device.	Addis (1972)* Bezdel (1970)** Braunstein & Anderson (1959)** Chapanis (1975, 1976)* Turn (1974)
Hand printing for optical character recognition (or for subsequent entry by typist)	The constrained hand printing required for OCR input results in low input rates, and sometimes high recognition error rates as well. Although manual transcription of such data clearly cannot be avoided in many cases, the preponderance of evidence suggests that direct keyboard entry yields better performance than printing, with a little practice, even when users are not skilled typists. Some error and input rate data on handprinting exist, along with some information about the effect of various printing constraints on input performance.	Apsey (1976)** Devoe (1967)** Masterson & Hirsch (1962)** Smith (1967)** Strub (1971)**
Mark sensing	As with hand printing, this separate transcription results in lower input rates than does practiced, but unskilled, typing. Some error and input rate data exist. May be slightly faster than constrained hand printing.	Devoe (1967)** Kulp & Kulp (1972)**
Punched cards	Keypunching performance differs significantly from ordinary typing because of differences in both the machine and the typical data to be keyed. Some reasonably good data exist on keypunch timing and error rates.	Neal (1977)**
Touch-tone telephone	Several studies suggest that the touch-tone telephone is a satisfactory device for occasional use as a computer terminal, even by naive computer users. It seems clear, though, that it is not a satisfactory device for prolonged interaction or for significant amounts of non-numeric input.	Miller (1974)** Smith & Goodwin (1970) Witten & Madams (1977)

Figure 21. Input Device Types (Concluded)

In most cases, the basic input task dictates that one particular class of input device will be used. For example, a requirement for the input of text (case 1), in significant quantities, usually implies an alphabetic keyboard. Input rates available by selecting letters from a displayed list, using a lightpen, trackball, or whatever, would likely be unacceptable unless the typing component of the user's task is very small. In fairness, though, there is little empirical data on this (one study, by Morrill et al, 1968, was not methodologically adequate to resolve this issue). For occasional typing, the use of a touchpanel with a displayed keyboard might work, but there are serious problems of proprioceptive feedback here, and no known research to support such a design decision. Under some circumstances, constrained hand printing for input by optical character recognition may be acceptable, but input rates using this method are often lower than those achievable by even unskilled typing (Devoe, 1967; Masterson & Hirsch, 1962).

Similarly, the requirement for positional input -- whether discrete (case 4) or continuous (case 5) -- usually implies the use of an explicitly positional (or "graphical") input device. There are several such devices, including the lightpen, joystick, trackball, mouse, graphical input tablet, and touch panel. Selection among these devices depends on a more detailed analysis of the user's tasks.

When the user's task is the selection of individual words or characters from a displayed list or text (case 3), either keyboard or graphical input devices may be appropriate, and most of the comparative experiments have been concerned with this situation. Earl and Goff (1965) compared "type-in" and (simulated) "point-in" input methods for selecting a word from a displayed list, as in command or operand selection. Although experienced typists performed better with type-in input, everyone else exhibited greater accuracy, and perhaps greater speed, when point-in input was used. This performance advantage probably exists only when the alternative list is relatively short and contains readily discriminable items. However, a properly designed command language should have those properties anyway (see previous section on command languages). Ramsey (unpublished)

has also found a well-designed lightpen input to be both faster and more accurate than function keys for very simple (verb-noun-adjective) command construction.

For selecting particular words or characters from a text display, as in text editing, a lightpen appears to be much faster than keyboard cursor controls (Goodwin, 1975), and a mouse appears to be both faster and more accurate than both ordinary cursor control keys and "text" keys which advance to the next character, word, line, or paragraph (Card et al, 1977). Among the graphical input devices investigated for this situation, the mouse and, next, the lightpen, appear preferable to a rate-controlled joystick, graphical input tablet, and knee control (Card et al, 1977; English et al, 1967). The trackball may be a very viable candidate here, but we are unaware of any comparative study involving its use for text selection. A touch panel might also be useable for selection of whole words, although individual character selection is probably beyond its resolution limits. No useable empirical data were found pertaining to the use of touch panels for any task.

The survey found few empirical studies of continuous positional input, such as occurs in freehand drawing. Herot (1976) and Negroponte (1975) provide a very interesting discussion of software used for this task in conjunction with a graphical input tablet, but provide no empirical data. Irving et al (1976) found a trackball superior to both lightpen and joystick for drawing straight lines and circles, but did not investigate more sophisticated hand drawing.

It is important to consider all the input tasks of the user before selecting any input device or devices. In part, this is a simple matter of avoiding suboptimization and finding a cost-effective solution. If, for example, the user has considerable typing and a moderate amount of text editing to do, keyboard cursor controls or thumbwheel cursor controls mounted on the keyboard might be acceptable. Obviously, other factors enter into such a decision, such as user hourly cost, user discretion, system operating cost, basic terminal cost, etc. Consider, though, the similar case in which the

user must do considerable typing, a moderate amount of text editing, and some freehand drawing. If, in this case, a trackball or lightpen is used for the drawing, it should probably also be used for cursor control or text selection in the text editing task. It would be nice to be able to assert that a great deal of human performance data exist to support such tradeoff decisions, but in fact, there appear to be very limited data for input devices other than keyboards.

Another reason for considering all the user's input tasks is the performance effect of mode mixing. When the user must alternate modes of input, as by alternately typing and using a lightpen, the performance effects can be highly detrimental. For example, Earl and Goff (1965) found both point-in and unskilled type-in input to be more accurate and faster than any of several conditions in which some point-in and some type-in input were combined. It is important that mode mixing problems be considered in selecting input devices. It is common, for example, for a trackball, mouse, or joystick to be selected in preference to a lightpen because the user must also often use the keyboard for typing. Card et al (1977) provide a useful analysis of the temporal properties associated with a simple case of mode mixing, and they provide a simple quantitative formula (based on Fitts' Law) which accounts for observed user performance differences among several text selection devices on the basis of the known dynamics of human hand movement.

If mode mixing affects user performance, then it is also important to know the assumptions underlying comparative performance studies. For example, both Card et al (1977) and English et al (1967) began each trial with depression of the space bar by the user, using the same hand that would then be used to control the text-selection device. This experimental paradigm clearly simulates a case in which typing is alternated with text selection. It is easy to imagine a dialogue in which such mode mixing is unnecessary, as by using a lightpen for <u>both</u> command and operand selection. It is quite likely that performance with the lightpen would at least equal that using a mouse, when this paradigm is used. Other factors are relevant here, though, especially the fatigue associated with long-term use of the lightpen.

The minimization of such problems requires the designer to take a systems view of the user's input requirements, and to determine the overall set of input devices based on that view. This can lead to a very different solution than might result from isolated consideration of each input task in turn. For an advanced word processing application, Engelbart (1973) has even gone so far as to advocate the operation of a mouse with one hand, for text selection, while the other hand operates a 5-key chorded alphabetic keyboard for both command selection and text input. (A standard alphanumeric keyboard is also provided.) This obviously requires a highly trained user, and may not often be cost-effective, but it illustrates a solution developed explicitly to minimize mode mixing.

Based on such considerations as have been discussed in this section, it is clear that procedural guidelines can be developed for use by the designer in selecting a set of input devices for a particular system. It is also clear, though, that such guidelines may be somewhat speculative, since inadequate empirical performance data exist with respect to many of the input devices and input tasks. Still, a human factors analysis of those devices and tasks would provide a firmer basis for design decisions than is now available to most interactive system designers.

Once the designer has chosen a particular type or types of input device, it is desirable to support further design decisions dealing with the selection or design of the specific keyboard, lightpen, etc., which will be used in the system. Except for keyboards, few empirical data were found which might support these detailed design decisions. This does not necessarily imply that no such data exist. In many cases, studies of these devices have been conducted outside the context of interactive computer systems, or unpublished studies have been done by manufacturers. Nonetheless, the overall body of knowledge in the area of detailed design parameters of nonkeyboard interactive input devices is disappointing.

In the case of keyboards, a very extensive literature exists, and only survey studies and a few specialized papers were included in our survey. A good descriptive survey of the literature is Alden et al (1972), although this report contains little in the way of derived conclusions or guidelines. Klemmer (1971) also provides a good, though brief, review of this literature. The review in Stewart et al (1974) is more critically penetrating and is directed primarily at interactive computer systems, but also provides little in the way of guidelines. The best prescriptive review of data entry devices and techniques in general, and keyboards in particular, is Seibel (1972). The guidelines presented by Seibel are crisp, sound, and are usually well founded in empirical research.

Such guidelines should be augmented by a somewhat more procedural discussion of keyboard design issues frequently encountered in the design of interactive systems. Such issues include determination, placement and grouping of special function keys, color coding of keys, and a variety of computer-specific functions (e.g., "break") which do not appear on typewriter keyboards (see also Dolotta, 1970).

TERMINALS, WORK STATIONS, AND FACILITY LAYOUT

Although displays and input devices have been discussed separately, there are some remaining design decisions which concern the packaging of such devices into terminals, consoles, and whole facilities. The issues involved are primarily those of physical layout, and are precisely the areas in which existing non-computer-related human engineering methods and guidelines are most directly applicable. Design guidelines in these areas can be highly quantitative, and should probably be much like Van Cott and Kinkade's (1972) guidelines for single-user workstation design or Thomson's (1972) guidelines for multi-user workstations. Standard human engineering texts are also applicable, as are portions of existing military standards (e.g., MIL-STD-1472B). Dolotta (1970) suggests a reasonably good set of standards for a teletypewriter terminal, but there are actually

very few guidelines dealing explicitly with computer-related workstation design.

Most of the literature which deals with computer-related console and facility layout consists of case studies. This information can provide a useful format for guidelines which will be used by designers unsophisticated in human factors. Very few new principles are to be found in these case studies, however, even though most are quite good. It would appear that existing guidelines on facility and console layout are about as applicable to computer systems as in other types of person-machine systems. Among the case studies are LeCocq's (1977) description of a CRT terminal design project, Shackel's (1962) design of a large console, Van Arsdel's (1974) purely physical discussion of a large NORAD information system, Wood's (1976) example of a console layout, and Galitz and Laska's (1969, 1970) layout of a computer facility. Although Yllo's (1962) physiological analysis dealt with a keypunch workstation, the methodology used is still applicable. There is, in fact, emerging evidence that the design of modern data entry workstations contributes to both postural fatigue (Duncan & Ferguson, 1974) and visual fatigue (Oestberg, 1975, 1976).

The survey uncovered almost nothing in the way of empirical research on overall workstation design for specifically computer-related tasks and settings. We are aware of some proprietary research on this topic in the United States, but it is unlikely that the results will soon become public. There is also work going on in Europe which concerns the design of terminal workstations for secretarial and clerical personnel. One such study involves the determination of appropriate physical dimensions of the terminal, table surface, etc., variations in the inclination and distance of the CRT display from the operator, and the use of a special carrier for source documents (George Sowden, personal communication). Guidelines for such a workstation might be helpful. In the United States, at least, it is very common to see commercial terminals placed on wholely inadequate tables, with little thought given to

work surface, background illumination, or other environmental factors. If workstation guidelines were integrated into a general design guide for interactive systems, this problem might receive at least some attention.

CONCLUSIONS

In this survey, we have considered a very broad range of literature. Although resource limitations necessarily prevented the inclusion of all of the papers potentially relevant to the field of human factors in computer systems, we feel that a reasonably clear and accurate picture of the overall state of the art emerged from the study. At the same time, some conclusions emerged with respect to the feasibility, potential utility, and possible form and content of a human factors guide to interactive computer system design.

1. The existing literature relevant to this field is badly fragmented because of its foundation in several different disciplines, and because relevant empirical data include those derived from many studies not specifically dealing with computer systems. Much of this literature is outside of that normally considered by human factors personnel, and the vast majority is outside the range used by interactive system designers. There is a strong need for the development of integrated guidelines.

2. While there is a large body of empirical data relevant to such guidelines, there are many significant gaps. In particular, there is inadequate information to support the development of highly quantitative "reference handbook" - type guidelines, except within certain fairly limited subdomains.

3. Consideration of the problem-solving behavior and information needs of the interactive system designer leads us to believe that "reference handbook" guidelines would not truly satisfy

the need anyway. What is needed is a design guide which is largely procedural in nature and is organized around the design process employed by designers.

4. Despite the existing gaps in our knowledge, a design guide of this sort appears feasible. Such an approach is compatible with the presentation of human factors methods, as well as empirical data and specific recommendations. In such a presentation, general psychological knowledge can often be used to advantage, especially in areas in which empirical information is sparse. In areas in which specific recommendations are impossible, this approach can at least direct the designer's attention to relevant factors.

5. Guidelines associated with the early system design process (e.g., user requirements analysis) will necessarily emphasize methods to be employed by the designer. Later, when the design decisions are more concrete and detailed -- and concern areas in which more empirical research has been done -- the guidelines can be more specific and prescriptive.

6. Although it is feasible to construct a design guide for interactive systems in general, it may be better to develop them for restricted types of systems (e.g., message processing, or tactical information systems). User behavior, and thus, desirable system properties, tend to be highly task-specific. By concentrating on a restricted range of user tasks, it should be possible to make guidelines more prescriptive

and explicit, and to use more meaningful examples, as well.

It is hoped that this survey has provided the reader with a greater understanding of the state of the art in human factors in computer systems, and that it has laid a sound foundation for the future construction of design guidelines in this area.

REFERENCES

Ackoff, R.L. Management misinformation systems. Management Science, 1967, 14, 147-156.

Addis, T.R. Human behaviour in an interactive environment using a simple spoken word recognizer. International Journal of Man-Machine Studies, 1972, 4, 255-284.

Alden, D.G., Daniels, R.W., & Kanarick, A.F. Keyboard design and operation: A review of the major issues. Human Factors, 1972, 14, 275-293 (A very similar paper by the same authors is Technical Report 12180-FR1A, Honeywell Systems and Research Center, St. Paul, Minnesota, March 1970).

Allan, J.J., III. Man-computer synergism for decision making in the system design process (Technical Report 9). Ann Arbor, Michigan: University of Michigan, Concomp Project, June 1968. (NTIS No. AD 673136)

Allen, T.J. Information needs and uses. In C.A. Cuadra & A.W. Luke (Eds.), Annual review of information science and technology (Vol. 4). Chicago: Encyclopedia Britannica, Inc., 1969, 3-29.

Alter, S. Why is man-computer interaction important for decision support systems? Interfaces, February 1977, 7(2), 109-115.

Anderson, R.H., & Gillogly, J.J. Rand Intelligent Terminal Agent (RITA): Design philosophy (Report No. R-1809-ARP1). Santa Monica, California: Rand Corp., 1976.

Apsey, R.S. Human factors of constrained handprint for OCR. In Proceedings, IEEE International Conference on Cybernetics and Society, November 1976. New York: Institute of Electrical and Electronics Engineers, Inc., 1976, 466-470.

Back, H.B. What information dissemination studies imply concerning the design of on-line reference retrieval systems. Journal of the American Society for Information Science, 1972, 23, 156-163.

Bailey, R.W. Testing manual procedures in commputer-based business information systems. In Proceedings of the 16th annual meeting of the Human Factors Society. Santa Monica, California: Human Factors Society, 1972, 395-401.

Baker, C.A. Morris, D.F., & Steedman, W.C. Target recognition in complex displays. Human Factors, 1969, 2, 51-61.

Baker, J.D. Quantitative modeling of human performance in information systems. Ergonomics, 1970, 13, 645-664. (NTIS No. AD 746096)

Baker, J.D., & Goldstein, I. Batch vs sequential displays: Effects on human problem solving. Human Factors, 1966, 8, 225-235.

Baldwin, J.T., & Siklossy, L. An unobtrusive computer monitor for multi-step problem solving. International Journal of Man-Machine Studies, 1977, 9, 349-362.

Bell Telephone Laboratories, Inc. Human error in computer-based data processing systems. Holmdel, New Jersey: Bell Telephone Laboratories, Inc., Human Performance Technology Center, 1973.

Bennett C.A., Toward empirical, practicable, comprehensive task taxonomy. Human Factors, 1971, 13, 229-235.

Bennett, J.L. Spatial concepts as an organizing principle for interactive bibliographic search. In D.E. Walker (Ed.), Interactive bibliographic search: The user/computer interface. Montvale, NJ: AFIPS Press 1971, 67-82.

Bentley, T.J. Defining management s information needs. AFIPS Conference Proceedings, 1976, 45, 869-876.

Bertin, J. Semiologie graphique: Les diagrammes -- les reseaux -- les cartes. The Hague: Mouton, 1967.

Best, J.B. Modeling planned moves in the hobbits-orcs task (unpublished manuscript). Bloomington, Indiana Indiana University, Department of Psychology, September 1977.

Bezdel, W. Some problems in man-machine communication using speech. International Journal of Man-machine Studies, 1970, 2, 157-168.

Boehm, B.W. Seven, M.J., & Watson, R.A. Interactive problem-solving: An experimental study of "lockout" effects. AFIPS Conference Proceedings, 1971, 38, 205-210.

Boies, S.J. User behavior on an interactive computer system. IBM Systems Journal, 1974, 13, 2-18.

Boies, S.J. Unpublished study cited in Miller and Thomas, 1977, p. 512.

Boies, S.J., & Spiegel, M.F. A behavioral analysis of programming: On the use of interactive debugging facilities (Technical Report RC-4472). Yorktown Heights, New York: IBM Watson Research Center, August 1973. (NTIS No. AD 772127)

Boomer, H.R. Relative comprehensibility of pictorial information and printed words in proceduralized instructions. Human Factors, 1975, 17, 266-277.

Booth, T.L., Kaufman, H.M., Lamb, J., Levy, R.M., Reiss, R.A., Sholl, H.A., & Vogelli, G. Experimental investigations of man-machine processing of information (Vol. III) (Technical Report U417-68-098). Groton, Connecticut: General Dynamics Corp., October 1968. (NTIS No. AD 684838)

Borko, H. Utilization of on-line interactive displays (Report No. SP-2575). Santa Monica, California: System Development Corp., August 1966. (NTIS No. AD 640652)

Bowen, H.M., Andreassi, J.L., Truax, S. & Orlansky, J. Graphical display of multiparametric information (Report No. AMRL-TDR-62-115). Wright-Patterson AFB, Ohio: 6570th Aerospace Medical Research Laboratories, 1963.

Bowen, R.J. Feehrer, C.E., Nickerson, R.S., & Triggs, T.J. Computer-based displays as aids in the production of army tactical intelligence (Technical Paper 258). Arlington, Virginia: U.S. Army Research Institute for the Behavioral and Social Sciences, February 1975. (NTIS No. AD A007819)

Braunstein, M., & Anderson, N.W. A comparison of the speed and accuracy of reading aloud and key-punching digits. IEEE Transactions on Human Factors in Electronics, 1961, HFE-2, 56-57.

Brown, J.S. & Burton, R.R. Diagnostic models for procedural bugs in basic mathematical skills. Cognitive Science, 1978, 2, 155-192.

Brown, J.S., Burton, R.R., Bell, A.G., & Bobrow, R.J. Sophie. A sophisticated instructional environment (Technical Report AFHRL-TR-74-93). Brooks Air Force Base, Texas: Air Force Human Resources Laboratory, December 1974. (NTIS No. AD A010109)

Brown, R.V., Peterson, C.R., Shawcross, W.H., & Ulvila, J.W. Decision analysis as an element in an operational decision aiding system (Phase II) (Technical Report 75-13). McLean, Virginia: Decisions and Designs, Inc., November 1975. (NTIS No. AD A018109)

Bryan, G.E. JOSS: 20,000 hours at a console -- a statistical summary. AFIPS Conference Proceedings, 1967, 31, 769-777.

Bryden, J.E. Design considerations for computer driven CRT displays. Computer Design, March 1969, 8(3), 38-46.

Buckler, A.T. A review of the literature on the legibility of alphanumerics on electronic displays (Technical memorandum 16-77). Aberdeen Proving Ground, Maryland: U.S. Army Human Engineering Laboratory, May 1977. (NTIS No. AD A040625)

Bursky, P., Churchill, W.H., Lull, B.E., Wagstaff, E.G., & Prywes, N.S. A man-machine competitive game: A naval duel (Technical Report 68-34). Philadelphia, Pennsylvania: University of Pennsylvania, Moore School of Electrical Engineering, May 1968. (NTIS No. AD 671911)

Callan, J.R., Curran, L.E., & Lane, J.L. Visual search times for Navy tactical information displays. (Report No. NPRDC-TR-77-32). San Diego, California: Navy Personnel Research and Development Center, May 1977. (NTIS No. AD A040543)

Carbonell, J.R. On man-computer interaction: A model and some related issues (BBN Report No. 1593). Cambridge, Massachusetts: Bolt, Beranek and Newman, Inc., September 1967 (also reported more briefly in IEEE Transactions on System Science and Cybernetics, 1969, SSC-5, 16-26). (NTIS No. AD 666666)

Carbonell, J.R. Mixed-initiative man-computer instructional dialogues (Report No. 1971). Cambridge, Massachusetts: Bolt, Beranek and Newman, Inc., May 1970. (NTIS No. AD 707782)

Carbonell, J.R. Artificial intelligence and large interactive man computer systems. Cambridge, Massachusetts: Bolt, Beranek and Newman, Inc., July 1971. (NTIS No. AD 726441)

Card, S.K., English, W.K., & Burr, B.J. Evaluation of mouse, rate-controlled isometric joystick, step keys, and text keys for text selection on a CRT. (Technical Report), Palo Alto, California: Xerox Palo Alto Research Center, 1977.

Card, W.I., Nicholson, M., Crean, G.P., Watkinson, G., Evans, C.R., Wilson, J., & Russell, D. A comparison of doctor and computer interrogation of patients. International Journal of Bio-Medical Computing, 1974, 5, 175-187.

Carlisle, J.H. Interactive man-machine communication: 1971 Annual Report (Technical Report No. 51). New Haven, Connecticut: Yale University, Department of Administrative Sciences, March 1972. (NTIS No. AD 740101)

Carlisle, J.H. The use of protocol analysis to design man computer interaction programs. In J.H. Carlisle, Interactive man-machine communication: 1972 Annual Report (Technical Report No. 66). New Haven, Connecticut: Yale University, Department of Administrative Sciences, March 1973, 30-64. (NTIS No. AD 760010)

Carlisle, J.H. Man-computer interactive problem solving: Relationships between user characteristics and interface complexity (Doctoral dissertation, Yale University, 1974). (University Microfilms No. 74-25725: Similar paper: NTIS No. AD 786466)

Carlson, P., & Hodgson, T.J. An interactive heuristic approach for scheduling a multi-resource constrained system (Report No. 77-10). Gainesville, Florida: September 1977. (NTIS No. AD A045081)

Carter, L.F. An experiment on the design of tables and graphs used for presenting numerical data. Journal of Applied Psychology, 1947, 31, 640-650.

Carter, L.F. Relative effectiveness of presenting numerical data by use of tables and graphs (Technical Memo No. TSEAA 694-1). Wright-Patterson AFB, Ohio: Air Materiel Command, 1948.

Caruso, D.E. Tutorial programs for operation of on-line retrieval systems. Journal of Chemical Documentation, 1970, 10, 98-105.

Chapanis, A. On the allocation of functions between men and machines. Occupational Psychology, 1965, 39, 1-11. (NTIS No. AD 626311)

Chapanis, A. Interactive human communication. Scientific American, March 1975, 232(3), 36-42.

Chapanis, A. Interactive human communication: Some lessons learned from laboratory experiments. Paper presented at NATO Advanced Study Institute on Man-Computer Interaction, Mati, Greece, September 1976.

Chesler, L., & Turn, R. Some aspects of man-computer communication in active monitoring of automated checkout (Technical Report P-3522). Santa Monica, California: Rand Corp., March 1967. (NTIS No. AD 648553)

Christ, R.E. Review and analysis of color coding research for visual displays. Human Factors, 1975, 17, 542-570.

Clarke, D.C. Query formulation for on-line reference retrieval: Design considerations from the indexer/searcher viewpoint. Proceedings of the American Society for Information Science, 1970, 7, 83-86.

Cochrane, J.L., & Zeleny, M. Multiple criteria decision making. Columbia, South Carolina: University of South Carolina Press, 1973.

Codd, E.F. Seven steps to redezvous with the casual user (Technical Report RJ-1333). San Jose, California: IBM Research Laboratory, January 1974.

Coffee, J.L. A comparison of vertical and horizontal arrangements of alpha-numeric material. Human Factors, 1961, 3, 93-98.

Cohen, J., & Dinnerstein, A.J. Flash rate as a visual coding dimension for information. (Technical Report No. WADC 57-64). Wright-Patterson AFB, Ohio: Wright Air Development Center, May 1958. (NTIS No. AD 118018)

Collins, A., Warnock, E.H., & Passafiume, J.J. Analysis and synthesis of tutorial dialogues (Report 2789). Cambridge, Massachusetts: Bolt, Beranek and Newman, Inc., March 1974 (also in G. Bower (Ed.), The psychology of learning and motivation, 1975, 9, 49-87).

Connelly, E.M. Manned system performance as a function of display characteristics (Technical Report No. OTR-62-77-1). Vienna, Virginia: Omnemii, Inc., June 1977.

Crawford, B.M., Topmiller, D.A., & Kuck, G.A. Man-machine design considerations in satellite data management (Report No. AMRL-TR-77-13). Wright-Patterson AFB, Ohio; Air Force Systems Command, Aerospace Medical Research Laboratory, April 1977. (NTIS No. AD A041287)

Cropper, A.G. Personal communication cited in Shackel and Shipley, 1970.

Cropper, A.G., & Evans, S.J.W. Ergonomics and computer display design. The Computer Bulletin, July 1968, 12(3), 94-98.

Cross, N. Simulation of computer-aided design. In Proceedings of the International Symposium on Man-machine Systems, 8-12 September 1969 (Vol. 1) (IEEE Conference Record No. 69C58-MMS). New York: Institute of Electrical and Electronics Engineers, Inc., 1969.

Croxton, F.E., & Stein, H. Graphic comparisons by bars, squares, circles, and cubes. Journal of the American Statistical Association, 1932, 27, 54-60.

Cushman, R.H. TOFT: A method for electronic doodling and a first step towards the use of computers on ill-defined problems. In Proceedings of the 1972 International Conference on Cybernetics and Society. New York: Institute of Electrical and Electronics Engineers, Inc., 1972, 157-162.

Danchak, M.M. CRT displays for power plants. Instrumentation Technology, October 1976, 23(10), 29-36.

Davis, K.B., Weisbrod, R.L., Freedy, A., & Weltman, G. Adaptive computer aiding in dynamic decision processes: An experimental study of aiding effectiveness (Technical Report PTR-1016-75-50). Woodland Hills, California: Perceptronics, Inc., May 1975. (NTIS No. AD A014218)

Dever, J.J. Gathering system user data to assist in the design of information processing systems. Paper presented at meeting of the Human Factors Society, Los Angeles, October 1972 (Reprinted by Bell Telephone Laboratories, Holmdel, NJ; proceedings contain abstract only).

Devoe, D.B. Alternatives to handprinting in the manual entry of data. IEEE Transactions on Human Factors in Electronics, 1967, HFE-8, 21-32.

Dickinson, G.C. Statistical mapping and the presentation of statistics (2nd Ed.). New York: Grane-Russakle Co., 1973.

Dickson, G.W., Senn, J.A., & Chervany, N.L. Research in management information systems: The Minnesota experiments. Management Science, 1977, 23(9), 913-923.

Dill, A.B., & Gould, J.D. Flickerless regeneration rates for CRT displays as a function of scan order and phosphor persistence. Human Factors, 1970, 12, 465-471.

Dolotta, T.A. Functional specifications for typewriter-like time-sharing terminals. Computing Surveys, 1970, 2, 5-31.

Dorris, A.L., Sadosky, T.L., & Connolly, T. Varying data and information in a decision making task. Ergonomics, 1977, 20, 643-649.

Doutriaux, J. Human-computer process control: Better training and better performance for the printing industry. In Proceedings of the 17th annual meeting of the Human Factors Society. Santa Monica, California: Human Factors Society, 1973, 111-115.

Duncan, J., & Ferguson, D. Keyboard operating posture and symptoms in operating. Ergonomics, 1974, 17, 651-662.

Durding, B.M., Becker, C.A., & Gould, J.D. Data organization (Research Report RC-4956). Yorktown Heights, New York: IBM Watson Research Center, July 1974.

Earl, W.K., & Goff, J.D. Comparison of two data entry methods. Perceptual and Motor Skills, 1965, 20, 369-384.

Eason, K.D. A task-tool analysis of manager-computer interaction. Paper presented at NATO Advanced Study Institute on Man-Computer Interaction, Mati, Greece, September 1976 (Reprinted by Department of Human Sciences, University of Technology, Loughborough, Leicestershire, England).

Eason, K. Human relationships and user involvement in systems design. Computer Management, May 1977, 10-12.

Eason, K.D., Damodaran, L., & Stewart, T.F.M. Interface problems in man-computer interaction. In E. Mumford & H. Sackman (Eds.), Human choice and computers. Amsterdam: North-Holland, 1975, 91-105.

Engel, S.E., & Granda, R.E. Guidelines for man/display interfaces (Technical Report TR 00.27200). Poughkeepsie, New York: IBM Poughkeepsie Laboratory, December 1975.

Engelbart, D.C. Design considerations for knowledge workshop terminals. AFIPS Conference Proceedings, 1973, 42, 221-227.

English, W.K., Engelbart, D.C., & Berman, M.L. Display-selection techniques for text manipulation. IRE Transactions on Human Factors in Electronics, 1967, HFE-8, 5-15.

Evans, C.R. What I have learnt about computer communication and what computers have taught me about human communication. Paper presented at NATO Advanced Study Institute on Man-Computer Interaction, Mati, Greece, September 1976 (Reprinted by Computer Science Division, National Physical Laboratory, Teddington, Middlesex, England).

Evans, J.A. A methodology for command information system analysis. In D.E. Walker (Ed.), Information system science and technology: Papers prepared for the third congress. Washington, D.C.: Thompson, 1967, 251-270.

Fetter, R.B., & Carlisle, J.H. Man-computer interaction in a decision-making environment (Report No. 1). New Haven, Connecticut: Yale University, Department of Administrative Sciences, March 1971. (NTIS No. AD 722336)

Feyock, S. Transition diagram-based CAI/HELP systems. International Journal of Man-Machine Studies, 1977, 9, 399-413.

Finkelman, J.M. Information processing load as a human factors criterion for computer systems design. In R.E. Granda & J.M. Finkelman (Eds.), The role of human factors in computers (Proceedings of a symposium co-sponsored by the Metropolitan Chapter of the Human Factors Society and Baruch College, City University of New York). New York: Human Factors Society, Metropolitan Chapter, 1976, 1-6.

Fitts, P.M., & Jones, R.E. Analysis of factors contributing to 460 "pilot error" experiences in operating aircraft controls. In H. W. Sinaiko (ed.), Selected papers on human factors in the design and use of control systems. New York: Dover, 1961, 332-358.

Flanagan, J.C. The critical incident technique. Psychological Bulletin, 1954, 51, 327-358.

Foley, J.D. An approach to the optimum design of computer graphics systems. Communications of the ACM, 1971, 14, 380-390.

Foley, J.D., & Wallace, V.L. The art of natural graphic man-machine conversation. Proceedings of the IEEE, 1974, 62, 462-471.

Ford, W.R. Self-limited and unlimited word usage during problem solving in two telecommunication modes (Technical Report No. 7). Baltimore, Maryland: Johns Hopkins University, Department of Psychology, April 1977.

Franklin, J., & Dean, E. Some expected and not so expected reactions to a computer-aided design with interactive graphics (CANDIG) system. SID Journal, May-June 1974, pp. 5-6; 8; 11-13.

Freedy, A., Davis, K.B., Steeb, R., Samet, M.G., & Gardiner, F.C. Adaptive computer aiding in dynamic decision processes: Methodology, evaluation, and applications (Technical Report PFTR-1016-76-8/30). Woodland Hills, California: Perceptronics, Inc., August 1976.

Freedy, A., Weltman, G., & Lyman, J. Interactive aspects of a man/learning system control team. In Proceedings of the 1972 International Conference on Cybernetics and Society. New York: Institute of Electrical and Electronics Engineers, Inc., 1972, 135-140.

Gagliardi, U.O., Hussey, R.A., Kaplan, I.T., & Matteis, R.J. Man-computer interactions in idealized tactical problem solving (Final Report, Contract NONR-3602(00)). Darien, Conecticut: Dunlap and Associates, Inc., May 1965. (NTIS No. AD 618707).

Galitz, W.O., & Laska, T.J. Computer system peripherals and the operator. Computer Design, August 1969, 8(8), 52-56.

Galitz, W.O., & Laska, T.J. The computer operator and his environment. Human Factors, 1970, 12, 563-573.

Gardner, J.A., & Soliday, S.M. Human factors in digital telecommunications. Human Factors, 1974, 16, 146-153.

Giddings, B.J. Alpha-numerics for raster displays. Ergonomics, 1972, 15, 65-72.

Gilb, T., & Weinberg, G.M. Humanized input: Techniques for reliable keyed input. Cambridge, Massachusetts: Winthrop, 1977.

Gold, M.M. A methodology for evaluating time-shared computer system usage. Unpublished doctoral dissertation, Massachusetts Institute of Technology, Cambridge, Massachusetts, June 1967.

Goodstein, L.P. A process instrumentation for man-machine communication studies. In Proceedings of the International Symposium on Man-Machine Systems, 8-12 September 1969 (Vol. 4) (IEEE Conference Record No. 69C58-MMS). New York: Institute of Electrical and Electronics Engineers, 1969.

Goodwin, N.C. Intro: In which a smart terminal teaches its own use (Technical Paper MTP-150). Beford, Massachusetts: Mitre Corp., March 1974.

Goodwin, N.C. Cursor positioning on an electronic display using lightpen, lightgun, or keyboard for three basic tasks. Human Factors, 1975, 17, 289-295.

Gould, J.D. Visual factors in the design of computer-controlled CRT displays. Human Factors, 1968, 10, 359-375.

Gould, J.D., & Ascher, R.N. Use of an IQF-like query language by non-programmers (Research Report No. RC-5279). Yorktown Heights, New York: IBM Watson Research Center, February 1975.

Gould, J.D., Lewis, C., & Becker, C.A. Writing and following procedural, descriptive, and restricted syntax language instructions (Research Report No. RC-5943). Yorktown Heights, New York: IBM Watson Research Center, April 1976.

Gould, J.D., & Makous, W.L. Vision and lasers: Human factors of laser displays. Information Display, November/December 1968.

Granda, R.E. Some considerations in defining the role of human factors in computers. In R.E. Granda & J.M. Finkelman (Eds.), The role of human factors in computers: Proceedings of a symposium co-sponsored by the Metropolitan Chapter of the Human Factors Society and Baruch College, City University of New York. New York: Human Factors Society, Metropolitan Chapter, 1977, 130-139.

Granda, T.M. An evaluation of visual search behavior on a cathode ray tube utilizing the window technique (Technical Paper 283). Alexandria, Virginia: U.S. Army Research Institue for the Behavioral and Social Sciences, February 1978. (NTIS No. AD A053352)

Green, B.F. The time required to search for numbers on large visual displays (Technical Report 36). Lexington, Massachusetts. MIT Lincoln Laboratory, 1953.

Green, J.S., V. GRINS, an on-line structure for the negotiation of inquiries (Studies in the Man-System Interface in Libraries Report No. 4). Bethlehem, Pennsylvania: Lehigh University, September 1967. (NTIS No. AD 660089)

Greeno, J.G. The structure of memory and the process of solving problems. In R.L. Solso (Ed.), Contemporary issues in cognitive psychology. Washington, D.C.: Winston, 1973.

Greenwald-Katz, G. Computers in architecture. AFIPS Conference Proceedings, 1976, 45, 315-320.

Grether, W.F., & Baker, C.A. Visual presentation of information. In H.P. Van Cott & R.G. Kinkade (Eds.), Human engineering guide to equipment design (Rev. Ed.). Washington, D.C.: U.S. Government Printing Office, 1972, 41-121.

Griffin, J.D.A. Experiences in displaying sequence-of-events information. In Proceedings, 3rd Man-Computer Communications Seminar. Ottawa, Ontario, Canada: National Research Council of Canada, 1973, 28.1-28.6.

Grignetti, M.C., Gould, L., Hausmann, C.L., Bell, A.G., Harris, G., & Passafiume, J. Mixed-initiative tutorial system to aid users of the On-Line System (NLS) (Technical Report ESD-TR-75-58). Hanscom Air Force Base, Bedford, Massachusetts: Electronic Systems Division, November 1974.

Grignetti, M.C., Miller, D.C., Nickerson, R.S., & Pew, R.W. Information processing models and computer aids for human performance: Task 2: Human-computer interaction models (Report No. AFOSR-TR-71-2845). Cambridge, Massachusetts: Bolt, Beranek and Newman, Inc., June 1971. (NTIS No. AD 732913)

Guttman, H.E., & Anderson, D.A. Studies comparing effectiveness of three-dimensional versus two-dimensional presentations (Report No. 1525-TR1). Minneapolis, Minnesota: Minneapolis-Honeywell Regulator Company, Military Products Group, November 1962.

Haeusing, M. Color coding of information on electronic displays. In Proceedings, 6th Congress of the International Ergonomics Association. Santa Monica, California: Human Factors Society, 1976, 210-217.

Halstead, M.H. Language level, a missing concept in information theory. Performance Evaluation Review (ACM SIGME), March 1973, 2(1), 7-9.

Hammell, T.J., Pesch, A.J., & Lane, W.P. Decision-making performance measurement for a command and control training system. In Proceedings, Human Factors Society, 19th annual meeting. Santa Monica, California: Human Factors Society, 1975, 315-320.

Hanes, L.F. Human factors in international keyboard arrangement. In A. Chapanis (Ed.), Ethnic variables in human factors engineering. Baltimore: Johns Hopkins University Press, 1975, 189-206.

Hanes, R.M. The effect of instructional format on functional testing performance. Human Factors, 1969, 11, 181-188.

Hanes, R.M., & Gebhard, J.W. The computer's role in command decision. U.S. Naval Institute Proceedings, September 1966, 92(9), 60-68.

Haralambopoulos, G., & Nagy, G. Profile of a university computer user community. International Journal of Man-Machine Studies, 1977, 9, 287-313.

Heafner, J.F. Protocol analysis of man-computer languages: Design and preliminary findings (Report No. ISI/RR-75-34). Marina del Rey, California: University of Southern California, Information Sciences Institute, July 1975. (NTIS No. AD A013568)

Hemingway, J.C., & Erickson R.A. Relative effects of raster scan lines and image subtense on symbol legibility on television. Human Factors, 1969, 11, 331-338.

Herot, C.F. Graphical input through machine recognition of sketches. In Proceedings of the Third Annual Conference on Computer Graphics, Interactive Techniques, and Image Processing -- SIGGRAPH '76. Computer Graphics, 1976, 10(2), 97-102.

Hill, I.D. Wouldn't it be nice if we could write computer programs in ordinary English -- or would it? Computer Bulletin, June 1972, 16(6), 306-312.

Hinds, G.H. The accuracy of data preparation. Computer Bulletin, June 1960, 4, 7-9.

Hlady, A.M. A touch sensitive X-Y position encoder for computer input. AFIPS Conference Proceedings, 1969, 35, 545-551.

Hoisman, A.J., Hannah, L.D., & Scharf, E. An experimental investigation of intelligence information display parameters (Report No. RADC-TR-63-378). Griffiss Air Force Base, New York: Rome Air Development Center, September 1963. (NTIS No. AD 422076)

Hormann, A. Problem solving and learning by man-machine teams -- progress and planned investigations (Technical Report SDC-TM-2311/ 008/00). Santa Monica, California: System Development Corp., July 1967. (NTIS No. AD 676123)

Howell, W.C., & Gettys, C.F. Some principles for design of decision systems: A review of the final phase of research on a comand-control system simulation (Technical Report No. AMRL-TR-68-158). Wright-Patterson Air Force Base, Ohio: Aerospace Medical Research Laboratories, November 1968. (NTIS No. AD 684548)

Howell, W.C., Johnston, W.A., & Goldstein, I.L. Influence of stress variables on display design (Technical Report RADC-TR-66-42). Griffiss Air Force Base, New York: Rome Air Development Center, April 1966. (NTIS No. AD 481509)

Howell, W.C., & Tate, J.D. Influence of display, response, and response set factors upon the storage of spatial information in complex displays. Journal of Applied Psychology, 1966, 50, 73-80.

Huff, D. How to lie with statistics. New York: Norton, 1954.

Hultgren, G.V., & Knave, B. Discomfort glare and disturbances from light reflections in an office landscape with CRT display terminals. Applied Ergonomics, 1974, 5, 2-8.

Igersheim, R.H. Managerial response to an information system. AFIPS Conference Proceedings, 1976, 45, 877-882.

International Purdue Workshop in Industrial Computer Systems. Guidelines for the design of man/machine interfaces for process control. West Lafayette, Indiana: Purdue University, Purdue Laboratory for Applied Industrial Control, October 1975.

Irving, G.W., Horinek, J.J., Chan, P.Y., & Walsh, D.H. Experimental investigation of sketch model accuracy and usefulness in a simulated tactical decision aiding task (Technical Report No. 215-3). Santa Monica, California: Integrated Sciences Corp., May 1977.

Irving, G.W., Horinek, J.J., Walsh, D.H., & Chan, P.Y. ODA pilot study II: Selection of an interactive graphics control device for continuous subjective functions applications (Report No. 215-2). Santa Monica, California: Integrated Sciences Corp., April 1976.

Ivergard, T.B.K. Man-computer interaction in public systems. Paper presented at NATO Advanced Study Institute on Man-Computer Interaction, Mati, Greece, September 1976 (Reprinted by Ergolab, Stockholm, Sweden, as Report No. 1975:22).

Jacob, R.J.K. Egeth, H.E., & Bevan, W. The face as a data display. Human Factors, 1976, 18, 189-200.

Johnson, J.K. Touching data. Datamation, January 1977, 23(1), 70-72.

Jordan, N. Allocation of functions between man and machines in automated systems. Journal of Applied Psychology, 1963, 47, 161-165.

Kasik, D.J. Controlling user interaction. In Proceedings of the Third Annual Conference on Computer Graphics, Interactive Techniques and Image Processing: SIGGRAPH '76. Computer Graphics (ACM SIGGRAPH), 1976, 10(2), 109-115.

Kelly, M.J. Limited vocabulary natural language dialogue. In Proceedings, Human Factors Society, 19th Annual Meeting. Santa Monica, California: Human Factors Society, 1975, 296-300.

King, W.R., & Cleland, D.I. The design of management information systems: An information analysis approach. Management Science, 1975, 22, 286-297.

Klemmer, E.T. Keyboard entry. Applied Ergonomics, 1971, 2, 2-6.

Kriebel, C.H. The evaluation of management information systems (Report No. RR-226). Pittsburgh, Pennsylvania: Carnegie-Mellon University, Management Sciences Research Group, September 1970. (NTIS No. AD 723083)

Krolak, P., Felts, W., & Marble, G. A man-machine approach toward solving the traveling salesman problem. Communications of the ACM, 1971, 14, 327-334.

Kulp, R.A., & Kulp, M.J. A comparison of mark sensing and handprinting coding methods. In Proceedings of the Sixteenth Annual Meeting of the Human Factors Society. Santa Monica, California: Human Factors Society, 1972, 416-421.

Landis, D., Slivka, R.M., Jones, J.M., Harrison, S., & Silver, C.A. Evaluation of large scale visual displays (Technical Report No. RADC-TR-67-57). Griffiss AFB, Rome, New York: Rome Air Developmental Center, April 1967. (NTIS No. AD 651372)

Le Cocq, A.D. The design of a CRT computer terminal: A human factors dream or nightmare? In Proceedings of the 21st Annual Meeting of the Human Factors Society. Santa Monica, California: Human Factors Society, 1977, 164-166.

Lenorovitz, D.R., & Ramsey, H.R. A dialogue simulation tool for use in the design of interactive computer systems. In Proceedings of the 21st Annual Meeting of the Human Factors Society. Santa Monica, California: Human Factors Society, 1977, 95-99.

Levit, R.A., Heaton, B.J., & Alden, D.G. Development and application of decision aids for tactical control of battlefield operations: Decision support in a simulated tactical operations system (SIMTOS). Minneapolis, Minnesota: Honeywell Systems and Research Center, November 1975.

Lewis, R.A. Legibility of capital and lowercase computer printout. Journal of Applied Psychology, 1972, 56, 280-281.

Ling, R.F. A computer generated aid for cluster analysis. Communications of the ACM, 1973, 16, 355-361.

Linstone, H.A., & Turoff, M. (Eds.). The Delphi method: Techniques and applications. Reading, Massachusetts: Addison-Wesley, 1975.

Lucas, R.W. A study of patients' attitudes to computer interrogation. International Journal of Man-Machine Studies, 1977, 9, 69-86.

Lumsdaine, A.A., Hunt, E., Beach, L.R., Pagano, R.R., & Wells, H.H. Learning and optimization of cognitive capabilities (Technical Report AFOSR 70-2761TR). Seattle, Washington: University of Washington, Department of Psychology, June 1970. (NTIS No. AD 716459)

Lutz, R.R. Graphic presentation simplified. New York: Funk & Wagnall, 1949.

Maddox, M.E. Prediction of information transfer from computer-generated dot matrix displays. In Proceedings of the 21st Annual Meeting of the Human Factors Society. Santa Monica, California: Human Factors Society, 1977, 38-42.

Mann, W.C. Dialogue-based research in man-machine communication (Technical Report ISI RR-75-41). Marina del Rey, California: University of Southern California, Information Sciences Institute, November 1975.

Martin, J. Design of man-computer dialogues. Englewood Cliffs, New Jersey: Prentice-Hall, 1973.

Martin, T.H., Carlisle, J., & Treu, S. The user interface for interactive bibliographic searching: An analysis of the attitudes of nineteen information scientists. Journal of the American Society for Information Science, 1973, 24, 142-147.

Martin, T.H., & Parker, E.B. Designing for user acceptance of an interactive bibliographic search facility. In D.E. Walker (Ed.), Interactive bibliographic search: The user/computer interface. Montvale, New Jersey: AFIPS Press, 1971, 45-52.

Martin, W.L., & Chubb, G.P. Human engineering considerations in designing interactive graphic displays for logistics management. Paper presented at Meeting of the Human Factors Society, San Francisco, October 1970 (Reprinted by Aerospace Medical Research Laboratory, Wright-Patterson AFB, Ohio).

Masterson, J.L., & Hirsch, R.S. Machine recognition of constrained handwritten Arabic numbers. IRE Transactions on Human Factors in Electronics, 1962, HFE-3, 62-65.

Mayzner, M.S. Factors affecting information storage and retrieval in man. New York: New York University, 1966. (NTIS No. AD 486382)

Mayzner, M.S., & Gabriel, R.F. The effect of spatial organization of the stimulus on short-term retention. Journal of Psychology, 1964, 58, 17-21.

McAllister, C., & Bell, J.M. Human factors in the design of an interactive library system. Journal of the American Society for Information Science, 1971, 22, 96-104.

McKendry, J.M., Mace, D.J., & Baker, J.D. Implications of BESRL Research for displays in tactical information processing (Technical Research Report 1156). Arlington, Virginia: U.S. Army Behavioral Science Research Laboratory, January 1969. (NTIS No. AD 688581)

McKendry, J.M., Wilson, R.C., Mace, D.J., & Baker, J.D. Application of a method for determining information requirements in a field army (Technical Paper 247). Arlington, Virginia: U.S. Army Research Institute for the Behavioral and Social Sciences, August 1973. (NTIS No. AD 767262)

Meister, D. Behavioral foundations of system develoment. New York: Wiley, 1976.

Meister, D. Heresies: Brief essays on human factors. San Diego, California: Navy Personnel Research and Development Center, March 1977.

Meister, D., & Farr, D.E. The utilization of human factors information by designers. Human Factors, 1967, 9, 71-87.

Mezrich, J.J., Carlson, C.R., & Cohen, R.W. Image descriptors for displays (Technical Report No. PRRL-77-CR-7). Princeton, New Jersey: RCA Laboratories, February 1977. (NTIS No. AD A042492)

Michie, D., Fleming, J.G., & Oldfield, J.V. A comparison of heuristic, interactive, and unaided methods of solving a shortest-route problem. In D. Michie (Ed.), Machine Intelligence (Vol 3). New York: American Elsevier, 1968, 245-255.

Miller, L.A. Programming by non-programers. International Journal of Man-Machine Studies, 1974, 6, 237-260.

Miller, L.A., & Becker, C.A. Programming in natural English (Technical Report No. RC-5137). Yorktown Heights, New York: IBM Watson Research Center, November 1974. (NTIS No. AD A003923)

Miller, L.A., & Thomas, J.C., Jr. Behavioral issues in the use of interactive systems. International Journal of Man-Machine Studies, 1977, 9, 509-536.

Miller, R.B. Task description and analysis. In R.M. Gagne' (Ed.), Psychological principles in system development. New York: Holt, Rinehart and Winston, 1962, 187-228.

Miller, R.B. Psychology for a man-machine problem-solving system (Technical Report TR00.1246). Poughkeepsie, New York: IBM Corp., February 1965 (also in L. Thayer (Ed.), Communication theory and research: Proceedings of the First International Symposium. Springfield, Illinois: Charles C. Thomas, 1967, 310-347). (NTIS No. AD 640283)

Miller, R.B. Response time in man-computer conversational transactions. AFIPS Conference Proceedings, 1968, 33(pt. 1), 267-277.

Miller, R.B. Archetypes in man-computer problem solving. Ergonomics, 1969, 12, 559-581.

Mjosund, A. Toward a strategy for information needs analysis. Computers and Operations Research, 1975, 2, 39-47.

Montgomery, C.A. Is natural language an unnatural query language? Proceedings of the Association for Computing Machinery, 1972, 1075-1078.

Morefield, M.A., Wiesen, R.A., Grossberg, M., & Yntema, D.B. Initial experiments on the effects of system delay on on-line problem solving (Technical Report No. TN-1969-5). Cambridge, Massachusetts: Massachusetts Institute of Technology, Lincoln Laboratories, June 1969.

Morgan, C.T., Cook, J.S., Chapanis, A. & Lund, M.W. Human engineering guide to equipment design. New York: McGraw-Hill, 1963.

Morrill, C.S. Computer-aided instruction as part of a management information system. Human Factors, 1967, 9, 251-256.

Morrill, C.S., Goodwin, N.C., & Smith, S.L. User input mode and computer-aided instruction. Human Factors, 1968, 10, 225-232.

Myer, T.H. How well do people point? Grafacon Interface, 1968, 2 (Bolt, Beranek & Newman, Inc., Cambridge, Massachusetts.

Nawrocki, L.H. Alpha-numeric versus graphic displays in a problem-solving task (Technical Research Note 227). Arlington, Virginia: U.S. Army Behavior and Systems Research Laboratory, September 1972. (NTIS No. AD 748799)

Nawrocki, L.H., Strub, M.H., & Cecil, R.M. Error categorization and analysis in man-computer communication systems. IEEE Transactions on Reliability, 1973, R-22, 135-140.

Neal, A.S. Time intervals between keystrokes, records, and fields in data entry with skilled operators. Human Factors, 1977, 19, 163-170 (Also: Technical Report HFC-8. San Jose, California: IBM Corp., System Development Division, Human Factors Center, October 1974).

Negroponte, N. Idiosyncratic systems: Toward personal computers and understanding context. Cambridge, Massachusetts: Massachusetts Institute of Technology, Architecture Machine Group, June 1975.

Newell, A., & Simon, H.A. Human problem solving. Englewood Cliffs, New Jersey: Prentice-Hall, 1972.

Newman, J.C. The processing of two types of command statement: A contribution to cognitive ergonomics. Paper presented at NATO Advanced Study Institute on Man-Computer Interaction, Mati, Greece, September 1976.

Newman, J.C. User psychology: The experimental strategy. Newtownabbey, Northern Ireland: Ulster College, The Northern Ireland Polytechnic, 1977.

Newman, K.M., & Davis, A.R. Relative merits of spatial and alphabetic encoding of information for a visual display. Journal of Engineering Psychology, 1963, 1, 102-126.

Newman, W.M. Interactive graphical response and its effects on display system performance. In Proceedings of the International Symposium on Man-Machine Systems, 8-12 September 1969 (Vol. 4) (IEEE Conference Record No. 69C58-MMS). New York: Institute of Electrical and Electronics Engineers, Inc., 1969.

Newsted, P.R., & Wynne, B.E. Augmenting man's judgment with interactive computer systems. Internatinal Journal of Man-Machine Studies, 1976, 8, 29-59.

Nicholson, R.M., Wiggins, B.D., & Silver, C.A. An investigation into software structures for man/machine interactions. Arlington, Virginia: Analytics, Inc., February 1972. (NTIS No. AD 737266)

Nickerson, R.S., & Pew, R.W. Oblique steps toward the human-factors engineering of interactive computer systems (Report No. 2190). Cambridge, Massachusetts: Bolt, Beranek and Newman, Inc., July 1971 (also published as Appendix in M.C. Grignetti, D.C. Miller, R.S. Nickerson, & R.W. Pew, Information processing models and computer aids for human performance. Cambridge Massachusetts: Bolt, Beranek and Newman, Inc., June 1971). (NTIS No. AD 732913)

Noll, A.M. Man-machine tactile communication. Sid Journal, July/August 1972, 1(2), 5-11.

Norman, D.A., & Bobrow, D.G. On data-limited and resource-limited processes. Cognitive Psychology, 1975, 1, 44-64.

Obermayer, R.W. Accuracy and timeliness in large-scale data-entry subsystems. In Proceedings of the 21st Annual Meeting of the Human Factors Society. Santa Monica, California: Human Factors Society, 1977, 173-177.

Oestberg, O. CRTs pose health problems for operators. International Journal of Occupational Health and Safety, November-December 1975, 44(6), pp. 24-26; 52; 50; 46.

Oestberg, O. Office computerization in Sweden: Worker participation, workplace design considerations, and the reduction of visual strain. Paper presented at NATO Advanced Study Institute on Man-computer Interaction, Mati, Greece, September 1976 (Reprinted by Department of Human Work Sciences, University of Lulea, Lulea, Sweden).

Parker, E.B. Behavioral research in the development of a computer-based information system. In C.E. Nelson & D.K. Pollock (Eds.), Communication among scientists and engineers. Lexington, Massachusetts: Health and Co., 1970, 281-292.

Parsons, M.M. Man-machine system experiments. Baltimore: The Johns Hopkins Press, 1972.

Peace, D.M.S., & Easterby, R.S. The evaluation of user interaction with computer-based management information systems. Human Factors, 1973, 15, 163-177.

Peters, G.L., & Barbato, G.J. Information processing of dot matrix displays (Technical Report AFFDL-TR-76-82). Wright-Patterson Air Force Base, Ohio: Air Force Flight Dynamics Laboratory, October 1976. (NTIS No. AD A036668)

Pew, R.W., Baron, S., Feehrer, C.E., & Miller, D.C. Critical review and analysis of performance models applicable to man-machine systems evaluation (Report No. 3446). Cambridge, Massachusetts: Bolt, Beranek and Newman, Inc., March 1977. (NTIS No. AD A038597)

Pew, R.W., & Rollins, A.M. Dialog specification procedures (Rev. Ed.) (Technical Report No. 3129). Cambridge, Massachusetts: Bolt, Beranek and Newman, Inc., September 1975.

Pew, R.W., Rollins, A.M., & Wiliams, G.A. Generic man-computer dialogue specification: An alternative to dialogue specialists. In Proceedings, 6th Congress of the International Ergonomics Association. Santa Monica, California: Human Factors Society, 1976, 251-254.

Plath, W.J. Restricted English as a user language. Yorktown Heights, New York: IBM Watson Research Center, November 1972.

Poulton, E.C., and Brown, C.H. Rate of commprehension of an existing teleprinter output and of possible alternatives. Journal of Applied Psychology, 1968, 52, 16-21.

Price, J.R., et al. Construct validity of test items measuring acquisitions of information from line graphs. Journal of Educational Psychology, 1974, 66, 152-156.

Prince, M.D. Interactive graphics for computer-aided design. Reading, Massachusetts: Addison-Wesley, 1971.

Proctor, J.H. Normative exercising: An analytic and evaluative aid in system design. IEEE Transactions on Engineering Management, 1963, EM-10, 183-192.

Pulfer, J.K. Man-machine interaction in creative applications. International Journal of Man-Machine Studies, 1971, 3, 1-11.

Raichelson, G. On-line terminal systems in the military environment. In Proceedings of the 1972 International Conference on Cybernetics and Society. New York: Institute of Electrical and Electronics Engineers, Inc., 1972, 126-128.

Raitzer, G.A., Vanderheiden, G.C., & Holt, C.S. Interfacing computers for the physically handicapped: A review of international approaches. AFIPS Conference Proceedings, 1976, 45, 209-216.

Ramsey, H.R. A formal approach to the design of interactive scheduling and resource allocation systems. Paper presented at the Military Operations Research Society 36th Symposium, Quantico, Virginia, December 1975.

Ramsey, H.R., Atwood, M.E., & Kirshbaum, P.J. A critically annotated bibliography of the literature on human factors in computer systems (Technical Report No. SAI-78-070-DEN). Englewood, Colorado: Science Applications, Inc., May 1978.

Reisner Planning Sciences, 1972, 6, 95-123.

Reisner, P. Use of psychological experimentation as an aid to development of a query language. IEEE Transactions on Software Engineering, 1977, SE-3, 218-229.

Reisner, P., Boyce, R.F., & Chamberlin, D.D. Human factors evaluation of two data base query languages: Square and Sequel. AFIPS Conference Proceedings, 1975, 44, 447-452.

Ridsdale, B. The non-specialist user and the computer terminal. In Man-computer interaction (Proceedings, Conference on Man-computer Interaction, 2-4 September 1970) (Conference Publication No. 68). London, England: Institution of Electrical Engineers, 1970, 153-159.

Ringel, S., Baker, J.D., Strub, M.H., & Kensinger, L.L. Human factors research in command information processing systems: Summary of recent studies (Technical Report 1158). Arlington, Virginia: Army Behavioral Science Research Laboratory, July 1969. (NTIS No. AD 694347)

Ringel, S., & Hammer, C. Information assimilation from alphanumeric displays: Amount and density of information presented (Technical Report TRN141). Washington, D.C.: U.S. Army Personnel Research Office, 1964. (NTIS No. AD 601973)

Ringel, S., Vicino, F.L., & Andrews, R.S. Human factors research in command information processing systems (Technical Report 1145). Washington, D.C.: Army Personel Research Office, March 1966. (NTIS No. AD 634313)

Robins, J.E., Buffardi, L., & Ryan, T.G. Research on tactical military decision making: Application of a decision prediction concept in a SIMTOS environment (Technical Paper 246). Arlington, Virginia: U.S. Army Research Institute for the Behavioral and Social Sciences, March 1974. (NTIS No. AD 780812)

Rodriguez, H., Jr. Measuring user characteristics on the Multics system (Technical Report No. MIT/LCS/TM-89). Cambridge, Massachusetts: Massachusetts Institute of Technology, August 1977.

Rogers, C.A., Balintfy, J.L., & Goldman, A.M. Computers, physicians, and the diagnostic decision-making process. Human Factors, 1964, 6, 459-464.

Sacerdoti, E.D. A structure for plans and behavior (Technical Note 109). Menlo Park, California: Stanford Research Institute, August 1975.

Sayani, H.H. A basis for the design of operating system command and response languages. In ACM '76: Proceedings of the Annual Conference. New York: Association for Computing Machinery, 1976, 373-380.

Scanlan, L.A. Visual time compression: Spatial and temporal cues. Human Factors, 1975, 17, 337-345.

Schmid, C.F. Handbook of graphic presentation. New York: Ronald Press, 1954.

Schutz, H.G. An evaluation of formats for graphic trend displays: Experiment II. Human Factors, 1961, 3, 99-107.

Schutz, H.G. An evaluation of methods for presentation of graphic multiple trends: Experiment III. Human Factors, 1961, 3, 108-119.

Schwartz, J.L., & Taylor, E.F. Computer displays in the teaching of physics. AFIPS Conference Proceedings, 1968, 33, 1285-1292.

Seibel, R. Data entry devices and procedures. In H.P. Van Cott & R.G. Kinkade (eds.), Human engineering guide to equipment design (Revised Ed.). Washington, D.C.: U.S. Government Printing Office, 1972, 311-344.

Seven, M.J., Boehm, B.W., & Watson, R.A. A study of user behavior in problem-solving with an interactive computer (Report No. R-513-NASA). Santa Monica, California: Rand Corp., April 1971.

Shackel, B. Ergonomics in the design of a large digital computer console. Ergonomics, 1962, 6, 229-241.

Shackel, B., & Shipley, P. Man-computer interaction: A review of ergonomics literature and related research (Report No. DMP-3472). Hayes, Middlesex, England: EMI Electronics Ltd., February 1970.

Shemer, J.E., & Heying, D.W. Performance modeling and empirical measurements in a system designed for batch and time-sharing users. AFIPS Conference Proceedings, 1969, 35, 17-26.

Shubik, M. Games for society, business and war: Towards a theory of gaming. New York: Elsevier, 1975.

Siegel, A.I., & Fischl, M.A. Dimensions of visual information displays. Journal of Applied Psychology, 1971, 55, 470-476.

Siegel, A.I., Fischl, M.A., & MacPherson, D.H. A forced-choice instrument for evaluating visual information displays. Wayne, Pennsylvania: Applied Psychological Services, Inc., 1969. (NTIS No. AD 687182)

Siegel, A.I., Wolf, J.J., & Leahy, W.R. A digital simulation model of message handling in the Tactical Operations System: I. The model, its sensitivity, and user's manual (Research Memorandum 73-5). Arlington, Virginia: U.S. Army Research Institute for the Behavioral and Social Sciences, November 1973.

Simon, H.A. Reflections on time sharing from a user's point of view. Science Research Review, 1966, 43-51.

Simon, H.A. The sciences of the artificial. Cambridge, Massachusetts: M.I.T. Press, 1969.

Slack, W. Computer-based interviewing system dealing with nonverbal behavior as well as keyboard responses. Science, 1971, 171, 84-87.

Smith, H.T. Man-computer collaboration in the design process (Memo No. 59). Sheffield, England: Department of Psychology, MRC Social and Applied Psychology Unit, September 1974 (Also Paper presented at Computer Aided Design 74, Imperial College, September 1974).

Smith, H.T., & Crabtree, R.G. Interactive planning: A study of computer aiding in the execution of a simulated scheduling task. International Journal of Man-Machine Studies, 1975, 7, 213-231.

Smith, L.B. A comparison of batch processing and instant turnaround. Communications of the ACM, 1967, 10, 495-500.

Smith, R.L., Moe, G.L., & Klein, G.G. Develoment of graphic area displays for ASW attack management simulation (2 vols.) (Technical Report C-1003). Burbank, California: Ocean Technology, Inc., 1972. (NTIS Nos. AD 747708 and AD 747691)

Smith, S.L. Color coding and visual separability in information displays. Journal of Applied Psychology, 1963, 47, 358-364. (a)

Smith, S.L. Color-coded displays for data-processing systems. Electro-Technology, April 1963, 71(4), 63-69. (b)

Smith, S.L. Legibility of overprinted symbols in multicolored displays. Journal of Engineering Psychology, 1963, 2, 82-96. (c)

Smith, S.L., & Duggar, B.C. Do large shared displays facilitate group effort? Human Factors, 1965, 7, 237-244. (NTIS No. AD 633262)

Smith, S.L., & Goodwin, N.C. Computer-generated speech and man-computer interaction. Human Factors, 1970, 12, 215-223.

Smith, S.L., & Goodwin, N.C. Blink coding for information display. Human Factors, 1971, 13, 283-290.

Smith, S.L., & Goodwin, N.C. Another look at blinking displays. Human Factors, 1972, 14, 345-347.

Smith, S.L., & Goodwin, N.C. Check-reading as a measure of display legibility. In Proceedings of the 17th Annual Meeting of the Human Factors Society. Santa Monica, California: Human Factors Society, 1973, 21-24.

Solomon, C.J., & Papert, S. A case study of a young child doing turtle graphics in logo. AFIPS Conference Proceedings, 1976, 45, 1049-1056.

Steeb, R., & Freedy, A. Man-machine interaction in adaptive remote systems. In Proceedings, IEEE International Conference on Cybernetics and Society, November 1976. New York: Institute of Electrical and Electronics Engineers, Inc., 1976, 727-731.

Steele, K.A. CPM/PERT. In Proceedings, 2nd Man-Computer Communications Seminar. Ottawa, Canada: National Research Council of Canada, 1971, 81-84.

Stewart, T.F.M. Ergonomic aspects of man-computer problem solving. Applied Ergonomics, 1974, 5, 209-212.

Stewart, T.F.M. Displays and the software interface. Applied Ergonomics, 1976, 7, 137-146. (a)

Stewart, T.F.M. The specialist user. Paper presented at NATO Advanced Study Institute on Man-Computer Interaction, Mati, Greece, September 1976 (Reprinted by Department of Human Sciences, University of Technology, Loughborough, Leicestershire, England). (b)

Stewart, T.R., & Carter, J.E. Policy: An interactive computer program for externalizing, executing, and refining judgmental policy (Report No. 159). Boulder, Colorado: University of Colorado, Institute of Behavioral Science, June 1973.

Stocker, A.C. Displays, papers, and lighting: The visual system in command centers. Information Display, September/October 1964, 1(1), 16-26.

Strickland, R.G. A study of the possibilities of graphs as a means of instruction in the first four grades of elementary school. New York: Columbia University, 1938.

Strub, M.H. Evaluation of man-computer input techniques for military information systems (Technical Research Note 226). Arlington, Virginia: U.S. Army Behavior and Systems Research Laboratory, May 1971. (NTIS No. AD 730315)

Strub, M.H. Automated aids to on-line tactical data inputting (Technical Paper 262). Arlington, Virginia: U.S. Army Research Institute for the Behavioral and Social Sciences, February 1975. (NTIS No. AD A010350)

Strub, M.H., & Levit, R.A. Computer compatibility with decision style: Eddies in a bit stream. In Proceedings of the Human Factors Society 18th Annual Meeting. Santa Monica, California: Human Factors Society, 1974, 46-49.

Summit, R.K. Dialog and the user: An evaluation of the user interface with a major online retrieval system. In D.E. Walker (Ed.), Interactive bibligraphic search: The user/computer interface. Montvale, NJ: AFIPS Press, 1971, 83-94.

Taggart, W.M., Jr., & Tharp, M.O. A survey of information requirements analysis techniques. Computing Surveys, 1977, 9, 273-290.

Teichner, W.H., Christ, R.E., & Corso, G.M. Color research for visual displays (Report No. ONR-CR213-102-4F). Las Cruces, New Mexico: New Mexico State University, Department of Psychology, June 1977. (NTIS No. AD A043609)

Teitelman, W. "Do what I mean": The programmer's assistant. Computers and Automation, April 1972, 21(4), 8-11.

Theologus, C.G., Romashko, T., & Fleishman, E.A. Development of a taxonomy of human performance: A feasibility study of ability dimensions for classifying human tasks (Technical Report No. 5). Washington, D.C.: American Institutes for Research, January 1970. (JSAS MS. No. 321)

Thomas, J.C. Quantifiers and question-asking (Technical Report No. RC 5866). Yorktown Heights, New York: IBM Watson Research Center, February 1976.

Thomas, J.C., & Gould, J.D. A psychological study of query by example. AFIPS Conference Proceedings, 1975, 44, 439-445.

Thomson, R.M. Design of multi-man--machine work areas. In H.P. Van Cott & R.G. Kinkade (Eds.), Human engineering guide to equipment design (Rev. Ed.). Washington, D.C.: Government Printing Office, 1972, 419-466.

Thompson, D.A. Man-computer system: Toward balanced co-operation in intellectual activities. In Proceedings, International Symposium on Man-Machine Systems. (IEEE Conference Record No. 69C58-MMS) (Vol. 1). New York: Institute of Electrical and Electronics Engineers, September 1969.

Thompson, D.A. Interface design for an interactive information retrieval system: A literature survey and a research system description. Journal of the American Society for Information Science, 1971, 22, 361-373.

Topmiller, D.A. Mathematical models of human performance in man-machine systems (Technical Report AMRL-TR-68-22). Wright-Patterson AFB, Ohio: Aerospace Medical Research Laboratories, May 1968. (NTIS No. AD 673348)

Treu, S. Interactive comand language design based on required mental work. International Journal of Man-Machine Studies, 1975, 7, 135-149.

Turn, R. Speech as a man-computer communication channel (Report No. P-5120). Santa Monica, California: Rand Corp., January 1974.

Uber, G.T., Williams, P.E., Hisey, B.L., & Siekert, R.G. The organization and formatting of hierarchical displays for the on-line input of data. AFIPS Conference Proceedings, 1968, 33, 219-226.

Van Arsdel, J.H., Sr. Human factors role in equipment definition and facility layout for the Space Computational Center and the NORAD Computer System Improvement Program. In Proceedings of the 18th Annual Meeting of the Human Factors Society. Santa Monica, California: Human Factors Society, 1974, 634-640.

Van Cott, H.P., & Kinkade, R.G. Human simulation applied to the functional design of information systems. Human Factors, 1968, 10, 211-216.

Van Cott, H.P., & Kinkade, R.G. (Eds.). Design of individual workplaces. In H.P. Van Cott & R.G. Kinkade (Eds.), Human engineering guide to equipment design (Rev. Ed.). Washington, D.C.: Government Printing Office, 1972, 381-418.

Van Lehn, K., & Brown, J.S. Planning nets: A representation for formalizing analogies and semantic models of procedural skills, (Technical Report No. 1). Pittsburgh, Pennsylvania: University of Pittsburgh Learning Research and Development Center, December 1978.

Vandersluis, H.J. The man-machine interface for 1990 management information system displays (Technical Report AMCA-71-006). Washington, D.C.: Army Advanced Materiel Concepts Agency, September 1970. (NTIS No. AD 712998)

Vartabedian, A.G. Effects of parameters of symbol formation on legibility. Information Display, May 1970, 7(5), 23-26.

Vartabedian, A.G. The effects of letter size, case, and generation method on CRT display search time. Human Factors, 1971, 13, 363-368.

Vaughan, W.S., Jr. & Mavor, A.S. Behavioral characteristics of men in the performance of some decision-making task components. Ergonomics, 1971, 15, 267-277.

Vernon, M.D. Learning from graphical material. British Journal of Psychology, 1946, 36(3), 145-158.

Vernon, M.D. The visual presentation of factual data. British Journal of Educational Psychology, 1950, 20, 174-185.

Vernon, M.D. The use and value of graphical material with a written text. Occupational Psychology (London), 1952, 26, 96-100. (a)

Vernon, M.D. The use and value of graphical methods of presenting quantitative data. Occupational Psychology (London), 1952, 26, 22-34. (b)

Vernon, M.D. Presenting information in diagrams. Audio-Visual Communication Review, 1953, 1(3), 147-158.

Vicino, F.L., & Ringel, S. Decision making with updated graphic vs alpha-numeric information (Technical Research Note 178). Washington, D.C.: Army Personnel Research Office, November 1966. (NTIS No. AD 647623)

Vitz, P.C. Preference for different amounts of visual complexity. Behavioral Science, 1966, 11, 105-114.

Vlahos, P. The three-dimensional display: Its cues and techniques. Information Display, November-December 1965, 2(6), pp. 10; 13-20.

Wainer, H. The suspended rootogram and other visual displays: An empirical validation. The American Statistician, 1974, 28(4), 143-145.

Waksman, A. The interface problem in interactive systems. Behavior Research Methods and Instrumentation, 1974, 6, 235-236.

Wallace, V.L. The semantics of graphic input devices. In Proceedings, ACM Symposium on Graphic Languages, SIGPLAN Notices, June 1976, 11(6), 61-65.

Walther, G.H. The on-line user-computer interface: The effects of interface flexibility, experience, and terminal-type on user-satisfaction and performance. Colorado Springs, Colorado: U.S. Air Force Academy, Department of Astronautics and Computer Science, August 1973 (also Doctoral Dissertation, University of Texas, Austin, Texas, 1973). (NTIS No. AD 777314)

Walther, G.H., & O'Neil, H.F., Jr. On-line user-computer interface: The effects of interface flexibility, terminal type, and experience on performance. AFIPS Conference Proceedings, 1974, 43, 379-384.

Wargo, M.J., Kelley, C.R., Mitchell, M.D., & Prosin, D.J. Human operator response speed, frequency, and flexibility: A review analysis and device demonstration (Report No. CR-874). Washington, D.C.: NASA, 1967.

Washburne, J.N. An experimental study of various graphic, tabular, and textual methods of presenting quantitative material. Journal of Educational Psychology, 1927, 18(6), pp. 361-376; 465-476.

Wasserman, A.I. The design of 'idiot-proof' interactive programs. AFIPS Conference Proceedings, 1973, 42, M34-M38.

Watson, R.W. User interface design issues for a large interactive system. AFIPS Conference Proceedings, 1976, 45, 357-364.

Weinberg, G.M. The psychology of computer programming. New York: Van Nostrand Reinhold, 1971.

Weltman, G., Steeb, R., Freedy, A., Smith, M., & Weisbrod, R. Experimental study of man/machine interaction in adaptive computer aided control (Technical Report 73-10). Woodland Hills, California: Perceptronics, Inc., November 1973. (NTIS No. AD 775879)

Whitham, G.E. The determination of display screen size and resolution based on perceptual and information limitations. Information Display, July/August 1965, 2(4), 15-19.

Wilde, D.U. Iterative strategy design. American Documentation, 1969, 20, 90-91.

Williams, J.D. The effects of computer subsystem response time and response time variance on operator performance in an interactive computer system. Paper presented at meeting of the American Psychological Association, Chicago, 1975.

Witten, I.H., & Madams, P.H.C. The telephone enquiry service: A man-machine system using synthetic speech. International Journal of Man-Machine Studies, 1977, 9, 449-464.

Wood, J. The application of ergonomics in the design of a computer console for emergency services: A case study. In Proceedings, 6th Congress of the International Ergonomics Association. Santa Monica, California: Human Factors Society, 1976, 225-259.

Wood, R.C. Inferences from the UCSB on-line system for man-system interface design. In Proceedings of the 1972 International Conference on Cybernetics and Society. New York: Institute of Electrical and Electronics Engineers, Inc., 1972, 120-125.

Wright, P. Using tabular information. Ergonomics, 1968, 11, 331-343.

Wright, P., & Reid, F. Written information: Some alternatives to prose for expressing the outcomes of complex contingencies. Journal of Applied Psychology, 1973, 57, 160-166.

Wylie, C.D., Dick, R.A., & Mackie, R.R. Toward a methodology for man-machine function allocation in the automation of surveillance systems. Volume I: Summary (Technical Report 1722-F, Volume 1). Goleta, California: Human Factors Research, Inc., July 1975. (NTIS No. AD A017103)

Yllo, A. The bio-technology of card punching. Ergonomics, 1962, 5, 75-79.

Yntema, D.B., & Klem, L. Telling a computer how to evaluate multidimensional situations. IEEE Transactions on Human Factors in Electronics, 1965, HFE-6, 3-13.

Zeigler, B.P., & Sheridan, T.B. Human use of short-term memory in processing information on a console. IEEE Transactions on Human Factors in Electronics, 1965, HFE-6, 74-83.

DISTRIBUTION LIST

CDR Paul R. Chatelier
Office of the Deputy Under Secretary
of Defense
OUSDRE (E&LS)
Pentagon, Room 3D129
Washington, D.C. 20301

CAPT John Duncan
Office of the Secretary of Defense
(C3I)
Pentagon, Room 3C200
Washington, D.C. 20301

Director
Engineering Psychology Programs
Code 455
Office of Naval Research
800 North Quincy Street
Arlington, VA 22217 (5 cys)

Director
Information Systems Program
Code 437
Office of Naval Research
800 North Quincy Street
Arlington, VA 22217

Special Assistant for Marine
Corps Matters
Code 100M
Office of Naval Research
800 North Quincy Street
Arlington, VA 22217

Commanding Officer
ONR Branch Office
ATTN: Dr. J. Lester
Building 114, Section D
666 Summer Street
Boston, MA 02210

Commanding Officer
ONR Branch Office
ATTN: Dr. C. Davis
536 South Clark Street
Chicago, IL 60605

Commanding Officer
ONR Branch Office
ATTN: Mr. R. Lawson
1030 East Green Street
Pasadena, CA 91106

Office of Naval Research
Scientific Liaison Group
American Embassy, Room A-407
APO San Francisco. CA 96503

Director
Naval Research Laboratory
Technical Information Divisio
Code 2627
Washington, D.C. 20375 (6 cys)

Dr. Bruce Wald
Communications Sciences Division
Code 7500
Naval Research Laboratory
Washington, D.C. 20375

Dr. Robert G. Smith
Office of the Chief of Naval
Operations, OP987H
Personnel Logistics Plans
Washington, D.C. 20350

Dr. Jerry C. Lamb
Display Branch
Code TD112
Naval Underwater Systems Center
New London, CT 06320

Naval Training Equipment Center
ATTN: Technical Library
Orlando, FL 32813

Human Factors Department
Code N215
Naval Training Equipment Center
Orlando, FL 32813

Dr. Alfred F. Smode
Training Analysis and Evaluation
Group
Naval Training Equipment Center
Code N-00T
Orlando, FL 32813

Dr. Gary Poock
Operations Research Department
Naval Postgraduate School
Monterey, CA 93940

Commanding Officer
ONR Branch Office
ATTN: Dr. E. Gloye
1030 East Green Street
Pasadena, CA 91106

Dean of Research Administration
Naval Postgraduate School
Monterey, CA 93940

Mr. Paul Heckman
Naval Ocean Systems Center
San Diego, CA 92152

Mr. Warren Lewis
Human Engineering Branch
Code 8231
Naval Ocean Systems Center
San Diego, CA 92152

Dr. Ross L. Pepper
Naval Ocean Systems Center
Hawaii Laboratory
P.O. Box 997
Kailua, HI 96734

Dr. A. L. Slafkosky
Scientific Advisor
Commandant of the Marine Corps
Code RD-1
Washington, D.C. 20380

HQS, U.S. Marine Corps
ATTN: CCA40 (MAJOR Pennell)
Washington, D.C. 20380

Commanding Officer
MCTSSA
Marine Corps Base
Camp Pendleton, CA 92055

Chief, C^3 Division
Development Center
MCDEC
Quantico, VA 22134

Mr. Arnold Rubinstein
Naval Material Command
NAVMAT 08T24
Washington, D.C. 20360

Mr. W. Greenert
Naval Material Command
NAVMAT 034
200 Stovall Street
Alexandria, VA 22332

Commander
Naval Air Systems Command
Human Factors Program
NAVAIR 340F
Washington, D.C. 20361

Commander
Naval Air Systems Command
Crew Station Design,
NAVAIR 5313
Washington, D.C. 20361

Mr. Phillip Andrews
Naval Sea Systems Command
NAVSEA 0341
Washington, D.C. 20362

Naval Sea Systems Command
Personnel & Training Analyses Office
NAVSEA 074C1
Washington, D.C. 20362

Commander
Naval Electronics System Command
Human Factors Engineering Branch
Code 4701
Washington, D.C. 20360

Dr. George Moeller
Human Factors Engineering Branch
Submarine Medical Research Lab
Naval Submarine Base
Groton, CT 06340

Navy Personnel Research and
Development Center
Manned Systems Design, Code 311
San Diego, CA 92152

Navy Personnel Research and
Development Center
Code 305
San Diego, CA 92152

Navy Personnel Research and
Development Center
Management Support Department
Code 210
San Diego, CA 92151

CDR P. M. Curran
Code 604
Human Factors Engineering Division
Naval Air Development Center
Warminster, PA 18974

Mr. Ronald A. Erickson
Human Factors Branch
Code 3175
Naval Weapons Center
China Lake, CA 93555

Human Factors Engineering Branch
Code 1226
Pacific Missile Test Center
Point Mugu, CA 93042

Dean of the Academic Departments
U.S. Naval Academy
Annapolis, MD 21402

Human Factors Section
Systems Engineering Test
Directorate
U.S. Naval Air Test Center
Patuxent River, MD 20670

Human Factor Engineering Branch
Naval Ship Research and Development
Center, Annapolis Division
Annapolis, MD 21402

LCDR W. Moroney
Code 55MP
Naval Postgraduate School
Monterey, CA 93940

Mr. Merlin Malehorn
Office of the Chief of Naval
Operations (OP 102X)
Washington, D.C. 20350

Mr. J. Barber
HQS, Department of the Army
DAPE-MBR
Washington, D.C. 20310

Dr. Joseph Zeidner
Technical Director
U.S. Army Research Institute
5001 Eisenhower Avenue
Alexandria, VA 22333

Director, Organizations and
Systems Research Laboratory
U.S. Army Research Institute
5001 Eisenhower Avenue
Alexandria, VA 22333

Dr. Edgar M. Johnson
Organizations and Systems Research
Laboratory
U.S. Army Research Institute
5001 Eisenhower Avenue
Alexandria, VA 22333

Technical Director
U.S. Army Human Engineering Labs
Aberdeen Proving Ground, MD 21005

U.S. Army Aeromedical Research Lab
ATTN: CPT Gerald P. Krueger
Ft. Rucker, AL 36362

ARI Field Unit-USAREUR
ATTN: Library
C/O ODCSPER
HQ USAREUR & 7th Army
APO New York 09403

U.S. Air Force Office of Scientific
Research
Life Sciences Directorate, NL
Bolling Air Force Base
Washington, D.C. 20332

Dr. Donald A. Topmiller
Chief, Systems Engineering Branch
Human Engineering Division
USAF AMRL/HES
Wright-Patterson AFB, OH 45433

Air University Library
Maxwell Air Force Base, AL 36112

Dr. Gordon Eckstrand
AFHRL/ASM
Wright-Patterson AFB, OH 45433

North East London Polytechnic
The Charles Myers Library
Livingstone Road
Stratford
London E15 2LJ
ENGLAND

Dr. Kenneth Gardner
Applied Psychology Unit
Admiralty Marine Technology
Establishment
Teddington, Middlesex TW11 OLN
ENGLAND

Director, Human Factors Wing
Defence & Civil Institute of
Environmental Medicine
Post Office Box 2000
Downsview, Ontario M3M 3B9
CANADA

Dr. A. D. Baddeley
Director, Applied Psychology Unit
Medical Research Council
15 Chaucer Road
Cambridge, CB2 2EF
ENGLAND

PROF Brian Shackel
University of Technology
Department of Human Sciences
Computer Human Factors Group
Loughborough
ENGLAND

Defense Documentation Center
Cameron Station, Bldg. 5
Alexandria, VA 22314 (12 cys)

Dr. Stephen J. Andriole
Director, Cybernetics Technology Office
Defense Advance Research Projects Agency
1400 Wilson Blvd.
Arlington, VA 22209

Dr. Stephen B. Weiner
Executive Office of the President
Office of Administration - Rm. 149
Washington, D.C. 20503

Dr. Robert R. Mackie
Human Factors Research, Inc.
5775 Dawson Avenue
Goleta, CA 93017

Human Resources Research Office
300 N. Washington Street
Alexandria, VA 22314

Dr. Jesse Orlansky
Institute for Defense Analyses
400 Army-Navy Drive
Arlington, VA 22202

Professor Judea Pearl
Engineering Systems Department
University of California-Los Angeles
405 Hilgard Avenue
Los Angeles, CA 90024

Dr. T. B. Sheridan
Department of Mechanical Engineering
Massachusetts Institute of Technology
Cambridge, MA 02139

Dr. Arthur I. Siegel
Applied Psychological Services, Inc.
404 East Lancaster Street
Wayne, PA 19087

Dr. Harry Snyder
Department of Industrial Engineering
Virginia Polytechnic Institute and State University
Blacksburg, VA 24061

Dr. J. A. Swets
Bolt, Beranek & Newman, Inc.
50 Moulton Street
Cambridge, MA 02138

Dr. Gershon Weltman
Perceptronics, Inc.
6271 Variel Avenue
Woodland Hills, CA 91364

Dr. Alphonse Chapanis
Department of Psychology
The Johns Hopkins University
Charles and 34th Streets
Baltimore, MD 21218

Dr. Michael M. Hammer
Massachusetts Institute of Technology
Laboratory for Computer Science
545 Technology Square
Cambridge, MA 02139

Dr. Bill Curtis
General Electric Company
Information Systems Programs
1755 Jefferson Davis Highway
Arlington, VA 22202

Journal Supplement Abstract Service
American Psychological Association
1200 17th Street, N.W.
Washington, D.C. 20036 (3 cys)

Mr. Edward M. Connelly
Performance Measurement Associates, Inc.
131 Park Street, NW.
Vienna, VA 22180

Dr. Edward R. Jones
Chief, Human Factors Engineering
McDonnell-Douglas Astronautics Company
St. Louis Division
Box 516
St. Louis, MO 63166